Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/110424

Improved Pattern Matching Algorithm

Deep Suman Dev[1] *and Dakshina Ranjan Kisku*[2,*]

[1] Department of Information Technology, Neotia Institute of Technology, Management and Science (NITMAS), Jhinga, Diamond Harbour Road, Amira, South 24 Parganas-743368, West Bengal, India
[2] Department of Computer Science and Engineering, National Institute of Technology Durgapur, Durgapur-713209, Burdwan, West Bengal, India

Received: 13 Mar. 2017, Revised: 28 May 2017, Accepted: 10 Jun. 2017
Published online: 1 Jul. 2017

Abstract: Pattern matching problem aims to search the most similar pattern or object by matching to an instance of that pattern in a scene image. In order to address the issue of finding an object in the target image efficiently, the most distinctive features are computed from the query pattern and need to be searched in the scene image. The scene image is logically divided into a number of candidate windows which are then to be matched with the query pattern. Due to repeated matching of the query pattern with local candidate windows, the pattern matching process requires a large amount of space in memory as well as it needs to be executed fast. Thus, pattern matching algorithms need to be memory efficient and as fast as possible. This paper makes an attempt to deal with these issues by presenting two effective pattern matching algorithms, namely, strip subtraction and strip division. The efficacy of the proposed pattern matching algorithms is tested on two databases, viz. a local database and MIT-CSAIL database containing random objects. The experimental results are proved to be computationally efficient ones while the proposed algorithms are compared with some existing algorithms possessing a uniform experimental setup.

Keywords: Pattern matching, Haar transform, Haar Projection Value, Strip sum, Integral image

1 Introduction

Pattern matching/template matching [1], [10], [19] is considered as a well-known and widely accepted research area in computer vision and pattern recognition. Pattern matching requires to use a convolution mask or template which is to be customized to specific features of the search image and perform the scanning operation of the entire image to find similar instances (local window) of that template in the searched image. In this paper, pattern or template can be understood as any shape of an object or some non-contiguous ones. These shapes or non-contiguous objects are then served as models to be matched to local window in the search image for similarity measurement. With the advancements of latest pattern matching technologies and development of novel algorithms, the pattern matching field has been emerged as a significant area of computer vision, which has many impressive and extensive applications in automotive industry, robot visions, object tracking and detection, motion estimation, road and vehicle tracking, human activity recognition, law enforcement, forensics, military, transport systems, etc. For example, pattern matching problems could be used for finding a hand-shape in particular orientation or finding a face in an image consists of human faces as well as objects of different shapes with a non-uniform background. Even it could be a problem to find a missing object or pattern in some industrial products while inspection of finishing products are accomplished in an assembly line, etc. In addition to these applications, pattern matching has a large number of fascinating applications in content-based image retrieval, object localization, image segmentation, boundary information extraction, block matching in motion estimation, image-based rendering, feature extraction and many more. In these applications, as priori information about the target pattern which is available in the form of deformable shape, texture, boundary information, etc. The task of pattern matching is to find the pattern in the image with the available priori information.

Broadly speaking, the pattern matching problem [1], [10], [19] is one of the efficient approaches of identifying, localizing and recognizing a template or a pattern within an image. The pattern is a 2D image fragment which is

* Corresponding author e-mail: drkisku@cse.nitdgp.ac.in

very small in size compare to search image and this pattern may represent any living or non-living object given in digital form in the search image. In video applications, a pattern may take the form of a 3D spatio-temporal fragment, representing a collection/group of 2-D patterns. To find a pattern in an image, the entire image is scanned and then the similarity between the pattern and the local window (block) at every pixel location is computed (block matching) for object matching. Typically, we are interested in finding a pattern in a given image and for that, we could extend the technique which finds patterns in a search image using strip sum technique [10], [13], [15]. However, this pattern matching technique is not much proved to be a computationally efficient one both in terms of taking space in memory and time required by the algorithm and therefore, overall performance gets slow down for large datasets. Although a good number of fast pattern matching algorithms is proposed as an alternative to full search algorithms exploited and discussed in [10], [13], [15] and they are used Haar like features for pattern representations and subsequently, Haar Projection Values (HPV) are extracted. Finally, matching of the input pattern with the window in the image using the image square sum technique is performed.

The rectangle sum and Haar like features are widely used in the applications like object detection [20], object localization and classification [8] and pattern matching [19], etc. In [20], a new machine learning approach has been described using integral image method and Haar like features to represent the image characteristics and adaboost is used for learning. This approach is capable of processing the image rapidly and it is being determined that detection rates are found to be high for visual object detection. In addition, it combines a set of classifiers to form a cascade which easily discards background images. In [8], the branch-and-bound approach has been used to maximize a large class of classifier functions efficiently on sub-images having good speed and for that, the classifiers which have this property are used for object localization and classification. In [19], a fast full-search equivalent pattern matching method has been proposed. The method is based on Lp-norm based dissimilarity functions like sum of absolute differences (*SAD*) and the sum of squared differences (*SSD*). Correlation (a measure of the degree to which two objects agree, not necessarily in actual value, but in general behaviour) and phase angle method has been used in [1] for pattern matching to recognize an object. To achieve higher computational speed, spatial correlation architecture has replaced with spectral correlation spectrum for logic utilization in FPGA [11]. In [7], a fast pattern matching algorithm which can handle arbitrary 2D affine transformations to minimize the sum of absolute differences (*SAD*) error, has been proposed. It uses a sublinear algorithm that randomly examines a small number of pixels and further the algorithm is accelerated by branch and bound scheme. A method to speed-up the pattern matching process has been proposed in [22] where candidate match locations are determined using a cascaded block-wise computation of integral image based binary test patterns. Further, traditional template matching is applied at the candidate match locations to determine the best overall match. A one-dimensional template matching algorithm was proposed in [4] which is considered as an alternative for 2-D full search block matching algorithms. The algorithm consists of three steps. The first step converts the 2-D images into 1-D by summing up the intensity values of the image in two directions, viz. horizontal and vertical directions. In the second step, the template matching is performed among 1-D vectors using the similarity function sum of square differences. Finally, the decision of acceptance or rejection is taken based on the value of similarity function. A quick technique for pattern matching using cross-correlation has been proposed in [6] where cross correlation has performed on odd or even signals samples only.

In this paper, strip subtraction and strip division based pattern matching techniques have been exploited. An extension of strip sum has been presented in the proposed work and both strip subtraction and strip division operators are used to calculate the rectangle sum on both the pattern and sliding window in the image. Three subtraction operations are needed to compute the rectangle sum and obtain the Haar projection value (HPV). Due to changes in basic arithmetic operations by extending the strip sum technique into strip subtraction and strip division, it reduces memory usage as well as it takes less time to find the pattern in the given image. The contributions are five-fold. (a) It provides solid mathematical foundations for strip subtraction and strip division techniques in both the horizontal and vertical directions. These techniques are known as horizontal strip subtraction, vertical strip subtraction, horizontal strip division and vertical strip division in this paper. (b) Essential mathematical derivations and computation of Orthogonal Haar Transform (OHT) are also provided. (c) Extensive experiments are conducted on two databases and results are presented for the techniques mentioned in (a). (d) The next contribution provides a common framework that exploits time and space complexity for both strip subtraction and strip division techniques and it is to be proved that both these techniques are memory and time efficient. (e) The last contribution concentrates on comparing the proposed strip subtraction and strip division pattern matching techniques with two existing techniques, namely, strip sum and integral image.

1.1 Problem Formulation

In order to find a pattern in the given image, a search process is initiated and tells whether a pattern of size $M_1 \times M_2$ is present in an image of size $N_1 \times N_2$ or not, where $M_1 < N_1$ and $M_2 < N_2$. The pattern is compared with all the candidate windows of equal sizes determined

from the image. Let us consider, the pattern and candidate windows are represented as vectors $\bar{X_t}$ and $\bar{X_w^j}$ respectively of dimension N, where $j = 0\ldots$(number of candidate windows (w) - 1) and $N = M_1 \times M_2$. Now, for example, if a 2×2 pattern is to be searched in an image of dimension 16×16, then we have $N = 4$ and $W = (16-2+1)^2 = 225$. Orthogonal Haar Transform (OHT) [13], [15] and subsequently, Fast Orthogonal Haar Transform (FOHT) [10] for pattern matching has been presented with the sum of squared differences (*SSD*) in [3], [5] which is used to measure the distance 'd' or dissimilarity between the vectors $\bar{X_t}$ and $\bar{X_w^j}$ correspond to the pattern and candidate windows and it is represented as $|\bar{X_t} - \bar{X_w^j}|^2$. Horizontal strip subtraction (*HSSub*), vertical strip subtraction (*VSSub*), horizontal strip division (*HSD*) and vertical strip division (*VSD*) techniques are applied to the pattern as well as on the candidate windows. The rectangle sums are calculated using these techniques and determine vectors which represent pattern and candidate windows. These vectors are then characterized by Orthogonal Haar transform (OHT) and Fast Orthogonal Haar Transform (FOHT) to obtain the Haar Projection Value (HPV). The Haar transformation that projects a vector $\bar{X} \in R^N$ onto a linear subspace having B orthogonal basis vectors $\bar{V_N^0}, \bar{V_N^1}, \cdots, \bar{V_N^{(B-1)}}$ is represented as a projection value vector $\bar{Y^(B)}$ of length B and having B number of projection values. Now, $\bar{Y^(B)} = V_N^B \bar{X} = [\bar{V_N^0}, \bar{V_N^1}, \cdots, \bar{V_N^{(B-1)}}]^t \bar{X}$ where t stands for matrix transposition, $\bar{X}$ is of length N, V_N^B is a matrix of size $B \times N$. Every candidate window is then compared with the pattern at a time to find the d (dissimilarity between the pattern and selected candidate window). d_{min} is determined from all d to find the threshold (T), which is then used to discriminate the matched and unmatched windows and when it is found that $d(\bar{X_t}, \bar{X_w^j}) \leq T$ or $|\bar{X_t} - \bar{X_w^j}| \leq T$ or $|V_N^B \bar{X_t} - V_N^B \bar{X_w^j}| \leq T$ (pattern and candidate windows are represented as vectors $\bar{X_t}$ and $\bar{X_w^j}$ respectively), then $X_w^j (j^{th}$ candidate window) is saved as '*matched candidate window*'.

1.2 Related Works

On object tracking and pattern matching, lots of work has been done so far and still, researches are going on to achieve desired results to be determined in complex environments. In this section, we have mentioned briefly a few similar pattern matching techniques which have shown advanced performance than earlier pattern matching algorithms. For example, the methods using incremental dissimilarity approximations (IDA) [19], low resolution pruning (LRP) [2], Walsh Hadamard Transform (WHT) [5], Gray-code kernels (GCK) [3] and Fast Walsh Hadamard Transform (FWHT) [14] are successfully applied for pattern matching. In pattern matching, one of the fundamental operations is to search a pattern within an image over sliding windows. To achieve this goal pattern matching algorithms have been designed and implemented. Among these techniques, the sum of absolute differences (*SAD*) [19], the sum of squared differences (*SSD*), L_p distance as dissimilarity, Hamming distance [17] are found to be very relevant to the proposed one. However, there exists a technique based on a computation of the rectangle sum [13], [15] using horizontal strip sum and vertical strip sum shows that only one addition is required to get one rectangle sum and only three subtraction operations are needed to calculate Haar Projection Value (HPV) [10]. To achieve speed-up in the pattern matching process, a method is discussed in [22] where candidate match locations are determined using a cascaded block-wise computation of integral image based binary test patterns. Thereafter, traditional template matching is applied at the candidate match locations to determine the best overall match. In [19], a fast full-search equivalent pattern matching method is exploited. The method is based on Lp-norm based dissimilarity functions like the sum of absolute differences (*SAD*) and the sum of squared differences (*SSD*).

The manuscript is organized as follows. Section 2 describes computation of Haar Projection Value (HPV) on applying Haar wavelets using strip subtraction and strip division operations. Next section discusses the basic strip sum operation which is used to derive the proposed techniques. Section 4 exploits strip subtraction and strip division operations which are to be computed using rectangle sum and further, it reports how Haar Projection Value (HPV) can be determined from these two proposed techniques. In addition, pattern matching in the context of strip subtraction and strip division operations is also presented. Section 5 takes a dig into analyse the time and space required by the proposed algorithms and this section also shows the number of arithmetic operations to be needed by algorithms. Experimental results and comparison of the proposed techniques with two existing techniques are presented in Section 6 and concluding remarks are made in the last section.

2 Extraction of Haar Projection Values (HPVs)

Wavelets are described as a set of non-linear basis functions and to project a function in terms of wavelets, the wavelet basis functions can be chosen. In the context of graphical interpretation, basis vectors look like waves, so they are known as '*wavelets*'. Haar wavelets are a sequence of 'square-shaped' functions.

The mother Haar wavelet function $\phi(t)$ can be described as

$$\phi(t) = \begin{cases} 1, & if \ \ 0 \leq t < \frac{1}{2} \\ -1, & if \ \ \frac{1}{2} \leq t < 1 \\ 0, & otherwise \end{cases} \quad (1)$$

Its scaling function $\phi'(t)$ is given by

$$\phi'(t) = \begin{cases} 1, & if \ \ 0 \le t < 1 \\ 0, & otherwise \end{cases} \tag{2}$$

From Equations (1) and (2), it can be said that the Haar matrix depends on the size of the image. An image of size $N = 2^n \times 2^n$, where N is a number of 2-D Haar wavelet basis functions (HWBF) are formed with orthogonal property from mother Haar wavelet function $\phi(t)$. Like for $n = 1$ and $N = 4$, four 2-D HWBF are to be formed and they are represented as vectors V_N^0, V_N^1, V_N^2 *and* V_N^3. Now, rows are to be concatenated when $N = 4$ and it is given by $V_N^0 = (1,1,1,1)^T$, $V_N^1 = (1,1,-1,-1)^T$, $V_N^2 = (1,-1,0,0)^T$, *and* $V_N^3 = (0,0,1,-1)^T$. Therefore, the Haar matrix or Haar wavelet basis function (HWBF) can be written as

$$H_{N=4} = (V_N^0, V_N^1, V_N^2, V_N^3)^T = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \tag{3}$$

Haar basis vector in Equation (3) can be constructed by applying dilation or squeezing and shifting operations shown in Figure 1. It squeezes the vector $< 1,1,1,1 >$ to $< 1,1,-1,-1 >$ and further, squeezes the vector $< 1,1,-1,-1 >$ to $< 1,-1,0,0 >$. The $< 1,1 >$ pair gets squeezed and becomes a single 1, and similarly the pair $< -1,-1 >$ becomes a single -1. Next, shift operation on the resultant basis vector is performed and obtain $< 0,0,1,-1 >$ which is our final basis vector. Figure 1 shows a 2×2 Haar wavelet basis functions (HWBF).

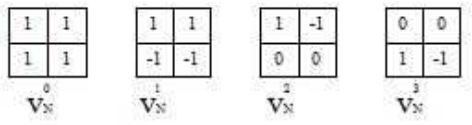


Fig. 1: Shows a 2×2 Haar Wavelet Basis Functions (HWBF)

Then magnitude of every HWBF in Equation (3) is calculated and each HWBF is divided row wise by its magnitude and obtain Haar normalized basis function (HNBF). The magnitudes of all vectors are given by

$$\text{Magnitude of } V_N^0 = \sqrt{1^2+1^2+1^2+1^2} = 2$$

$$\text{Magnitude of } V_N^1 = \sqrt{1^2+1^2+(-1)^2+(-1)^2} = 2$$

$$\text{Magnitude of } V_N^2 = \sqrt{1^2+(-1)^2+0^2+0^2} = \sqrt{2}$$

$$\text{Magnitude of } V_N^3 = \sqrt{0^2+0^2+1^2+(-1)^2} = \sqrt{2}$$

Then Haar normalized basis function (HNBF) is then given by

$$HNBF = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \tag{4}$$

HNBF determined in Equation (4) can be applied to pattern and all the candidate windows in the image and this results the Haar Projection Value (HPV) is also known as Haar Transform. Since HWBF have orthogonal property, therefore Haar transform in this case will be called Orthogonal Haar Transform (OHT) and the input image matrix can be represented as

$$I_M = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

To obtain the transformed 1-D Haar Matrix, each row of the input image matrix will have row-wise dot product with HNBF in Equation (4). For the first row in the input matrix $[a_{11}\ a_{12}\ a_{13}\ a_{14}]$ the representation would be done as follows

$$\begin{aligned} 1dh_{11} &= a_{11}/2 + a_{12}/2 + a_{13}/2 + a_{14}/2; \\ 1dh_{12} &= a_{11}/2 + a_{12}/2 - a_{13}/2 - a_{14}/2; \\ 1dh_{13} &= a_{11}/\sqrt{2} - a_{12}/\sqrt{2} + a_{13}\cdot 0 + a_{14}\cdot 0; \\ 1dh_{14} &= a_{11}\cdot 0 + a_{12}\cdot 0 + a_{13}/\sqrt{2} - a_{14}/\sqrt{2}; \end{aligned}$$

Therefore, $\lfloor 1dh_{11}\ 1dh_{12}\ 1dh_{13}\ 1dh_{14} \rfloor$ would be the 1-D Haar like representation of the first row $[a_{11}\ a_{12}\ a_{13}\ a_{14}]$ and this can be placed in the first row of transformed 1-D Haar Matrix. Similar operations are performed for all remaining rows of input image matrix. Then the transformed 1-D Haar matrix would be

$$\begin{bmatrix} 1dh_{11} & 1dh_{12} & 1dh_{13} & 1dh_{14} \\ 1dh_{21} & 1dh_{22} & 1dh_{23} & 1dh_{24} \\ 1dh_{31} & 1dh_{32} & 1dh_{33} & 1dh_{34} \\ 1dh_{41} & 1dh_{42} & 1dh_{43} & 1dh_{44} \end{bmatrix}$$

Now, the dot product is performed between each column of the above transformed 1-D Haar matrix and HNBF of Equation (4) and we obtain the transformed 2-D Haar matrix which is also known as Haar Projection Value (HPV) for the input matrix I_M. For the first column in transformed 1-D Haar matrix $[1dh_{11}\ 1dh_{21}\ 1dh_{31}\ 1dh_{41}]$ the representation would be done as follows

$$2dh_{11} = 1dh_{11}/2 + 1dh_{21}/2 + 1dh_{31}/2 + 1dh_{41}/2;$$
$$2dh_{12} = 1dh_{11}/2 + 1dh_{21}/2 - 1dh_{31}/2 - 1dh_{41}/2;$$
$$2dh_{13} = 1dh_{11}/\sqrt{2} - 1dh_{21}/\sqrt{2} + 1dh_{31}\cdot 0 + 1dh_{41}\cdot 0;$$
$$2dh_{14} = 1dh_{11}\cdot 0 + 1dh_{21}\cdot 0 + 1dh_{31}/\sqrt{2} - 1dh_{41}/\sqrt{2};$$

So, $[2dh_{11}\ 2dh_{21}\ 2dh_{31}\ 2dh_{41}]$ is 2-D Haar like representation of $[a_{11}\ a_{12}\ a_{13}\ a_{14}]$ is placed in the first row of transformed 2-D Haar matrix. Similarly, the operations would be performed for all remaining columns of transformed 1-D Haar matrix and the transformed 2-D Haar matrix can now be written as

$$\begin{bmatrix} 2dh_{11} & 2dh_{12} & 2dh_{13} & 2dh_{14} \\ 2dh_{21} & 2dh_{22} & 2dh_{23} & 2dh_{24} \\ 2dh_{31} & 2dh_{32} & 2dh_{33} & 2dh_{34} \\ 2dh_{41} & 2dh_{42} & 2dh_{43} & 2dh_{44} \end{bmatrix}$$

3 Overview of Strip Sum

Strip sum is considered as one of the rectangle sum techniques and this includes horizontal strip sum and vertical strip sum operations. Rectangle sum involves two edge operations, viz. one on the left and other at the top from the current location. In this section, basic building block of strip sum framework is exploited in both horizontal and vertical directions and this process initiates the development of strip subtraction and strip division characterized rectangle sum techniques.

3.1 Horizontal Strip Sum

Let $HSS_h(x,y)$ in Figure 2 be the horizontal strip sum with height *'h'* at (x,y) pixel. The horizontal strip sum can be defined as follows

$$HSS_h(x,y) = \begin{cases} I(x,y), & y = 0 \\ I(x,y+h) - I(x,y-1), & y > 0 \end{cases} \quad (5)$$

In Equation (5), for any point (x,y) and for $y = 0$, $HSS_h(x,y)$ would result only the image, otherwise, for $y > 0$, $HSS_h(x,y)$ would be given the result in horizontal extension. Rectangle sum using strip sum is calculated and the sum *Rect* area is given by

$$(P_1,P_2,N_1,N_2) = [I(P_1+N_1,P_2+N_2) \quad (6)$$
$$-I(P_1+N_1,P_2)] - [I(P_1,P_2+N_2) - I(P_1,P_2)]$$
$$= HSS(P_1+N_1,P_2,N_2) - HSS(P_1,P_2,N_2)$$

where $HSS(P_1 + N_1,P_2,N_2) = I(P_1+N_1,P_2+N_2) - I(P_1+N_1,P_2)$
and $HSS(P_1,P_2,N_2) = I(P_1,P_2+N_2) - I(P_1,P_2)]$

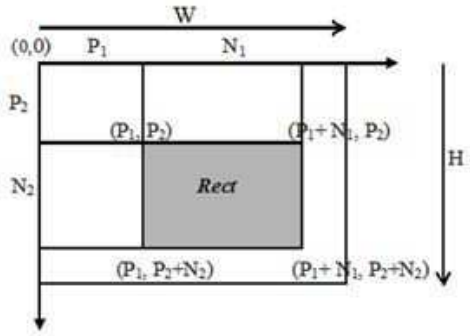


Fig. 2: An Illustration of Horizontal Strip Sum.

3.2 Vertical Strip Sum

Let, $VSS_w(x,y)$ in Figure 2 be the vertical strip sum with width $'w'$ at (x,y) pixel. Then vertical strip sum can be defined as follows

$$VSS_w(x,y) = \begin{cases} I(x,y), & x = 0 \\ I(x+w,y) - I(x-1,y), & x > 0 \end{cases} \quad (7)$$

Rectangle Sum using strip sum is calculated and the sum *Rect* area is given by

$$(P_1,P_2,N_1,N_2) = [I(P_1+N_1,P_2+N_2) \quad (8)$$
$$-I(P_1,P_2+N_2)] - [I(P_1+N_1,P_2) - I(P_1,P_2)]$$
$$= VSS(P_1,P_2+N_2,N_1) - VSS(P_1,P_2,N_1)$$

where $VSS(P_1,P_2 + N_2,N_1) = I(P_1+N_1,P_2+N_2) - I(P_1,P_2+N_2)$
and $VSS(P_1,P_2,N_1) = I(P_1+N_1,P_2) - I(P_1,P_2)$

4 Proposed Pattern Matching Process

In this section strip sum mechanism is extended with two basic arithmetic operations, namely, subtraction and division and they are to be used to achieve pattern matching in a given image. We call these two techniques as strip subtraction and strip division and they are being used to compute rectangle sum in both horizontal and vertical directions. From Equations (3) and (4) we compute Haar wavelets basis function (HWBF) and Haar normalized basis function (HNBF) using Haar wavelets, respectively. Further, these two basis functions are used to determine orthogonal Haar transform (OHT) and subsequently, we determine fast orthogonal Haar transform (FOHT). On both pattern and sliding windows rectangle sum using strip subtraction and strip division are computed and is characterized by orthogonal Haar transform. In Section 4.1 we compute rectangle sum using horizontal and vertical strip sum operations, and cumulative subtraction is performed on each pixel point

along two edges, viz. on the left and on the top. On the other hand, computation of rectangle sum using horizontal and vertical strip division operations are presented in Section 4.2. Section 4.3 discusses pattern matching with these two proposed techniques. In addition to comprehensive descriptions about the techniques, a solid mathematical foundation has been provided.

4.1 Horizontal and Vertical Strip Subtraction

To achieve horizontal and vertical strip subtraction, the cumulative subtraction operation is applied on the intensity values correspond to pixel points in the image. Suppose the gray value of a pixel (x,y) in an image is $g(x,y)$. The cumulative subtraction $CS(x,y)$ method cumulatively then subtracts values of all the gray levels in the rectangle area from origin the $(0,0)$ to (x,y).

To calculate cumulative subtraction of gray level for any location (x,y), the updated value (using cumulative subtraction) of the pixel $(x-1,y)$ and original pixel value of all the pixels from $(x,0)$ to $(x,y-1)$ are subtracted from the original image value at (x,y). The algorithm is given in Algorithm 1. The cumulative sum is given by the following equation

$$CS(x,y) = (g(x,y) - CS(x-1,y)) - \sum_{i=0}^{y-1} g(x,i) \quad (9)$$

Algorithm 1 Cumulative Subtraction (CS)

Let, $CS(0,0) = g(0,0)$ and d, c are two matrices having all zeros with size $h \times w$.
for $x = 0$ to $h-1$ **do**
  for $y = 0$ to $w-1$ **do**
    $b(x,y) = g(x,y)$
  end for
end for
for $x = 1$ to $h-1$ **do**
  $ComputeCS(x,0) = g(x,0) - CS(x-1,0)$
end for
for $y = 1$ to $w-1$ **do**
  for $k = 0$ to $y-1$ **do**
    $d(0,y) = b(0,k) - d(0,y)$
  end for
  $ComputeCS(0,y) = g(0,y) - d(0,y)$
end for
for $x = 1$ to $h-1$ **do**
  for $y = 0$ to $w-1$ **do**
    for $k = 0$ to $y-1$ **do**
      $c(x,y) = b(x,k) - c(x,y)$
    end for
    $CS(x,y) = (g(x,y) - CS(x-1,y)) - c(x,y)$
  end for

Horizontal strip subtraction $HSSub(P_1,P_2,N_2)$ in Figure 3 can now be defined as the sum of pixels $X(m,n)$ for $0 \le m \le P_1 - 1$, $P_2 \le n \le P_2 + N_2 - 1$ with the fixed height N_2 and any width N_1. $HSSub(P_1,P_2,N_2)$ can be computed by only one addition per pixel as follows using the cumulative subtraction defined in Equation (9)

$$HSSub(P_1,P_2,N_2) = \sum_{m}^{P_1-1} \sum_{n=P_2}^{P_2+N_2-1} X(m,n) = \cdots \quad (10)$$
$$\sum_{m=0}^{P_1-1} \sum_{n=0}^{P_2+N_2-1} X(m,n) - \sum_{m=0}^{P_1-1} \sum_{n=0}^{P_2-1} X(m,n) \cdots$$
$$CS(P_1,P_2+N_2) - CS(P_1,P_2)$$

To calculate the rectangle sum of *Rect* area $RS(P_1,P_2,N_1,N_2)$ and $RS(P_1,P_2,N_1',N_2)$, where $(N_1 \neq N_1')$ for $0 \le P_1, P_1+N_1 \le W, P_1+N_1' \le W; 0 \le P_2, P_2+N_2 \le H$, then rectangle sum in the context of horizontal strip sum is given by

$$\begin{aligned} RS \quad & (P_1,P_2,N_1,N_2) = [CS(P_1+N_1,P_2+N_2) - \cdots \\ CS \quad & (P_1+N_1,P_2)] - [CS(P_1,P_2+N_2) - \cdots \\ CS \quad & (P_1,P_2)] = HSSub(P_1+N_1,P_2,N_2) - \cdots \\ HSSub \; & (P_1,P_2,N_2) \end{aligned} \quad (11)$$

$$\begin{aligned} RS \quad & (P_1,P_2,N_1',N_2) = HSSub(P_1+N_1',P_2,N_2) - \cdots \\ HSSub \; & (P_1,P_2,N_2) \end{aligned} \quad (12)$$

Vertical Strip Subtraction $VSSub(P_1,P_2,N_1)$ in Figure 3 can be defined as the sum of pixels $X(m,n)$ for $P_1 \le m \le P_1+N_1-1$, $0 \le n \le P_2 - 1$ with the any height N_2 and fixed width N_1. $VSSub(P_1,P_2,N_1)$ can be computed by only one addition per pixel as follows using the cumulative subtraction defined in Equation (9)

$$VSSub \; (P_1,P_2,N_1) = \sum_{m=P_1}^{P_1+N_1-1} \sum_{n=0}^{P_2-1} X(m,n) = \cdots$$
$$\Sigma \quad \sum_{m=0}^{P_1+N_1-1} \sum_{n=0}^{P_2-1} X(m,n) - \sum_{m=0}^{P_1-1} \sum_{n=0}^{P_2-1} X(m,n) \cdots$$
$$CS \quad (P_1+N_1,P_2) - CS(P_1,P_2) \quad (13)$$

To calculate rectangle sum of *Rect* area $RS(P_1,P_2,N_1,N_2)$, where for $0 \le P_2$, $P_2+N_2 \le H$; $0 \le P_1$, $P_1+N_1 \le W$, we can calculate rectangle sum as follows

$$\begin{aligned} RS \quad & (P_1,P_2,N_1,N_2) = [CS(P_1+N_1,P_2+N_2) - \cdots \\ CS \quad & (P_1,P_2+N_2)] - [CS(P_1+N_1,P_2) - \cdots \\ CS \quad & (P_1,P_2)] = VSSub(P_1,P_2+N_2,N_1) - \cdots \\ VSSub \; & (P_1,P_2,N_1) \end{aligned} \quad (14)$$

For sliding N_2, where $N_2' > N_2$, we can write

$$RS \quad (P_1,P_2,N_1,N_2') = VSSub(P_1,P_2+N_2',N_1) - \cdots$$
$$VSSub\ (P_1,P_2,N_1) \tag{15}$$

With respect to the transformed image values on the use of Haar basis function, Haar Projection Values (HPVs) are calculated and from Equations (11) and (14), rectangle sum by horizontal strip subtraction ($HSSub$) and vertical strip subtraction ($VSSub$) being calculated.

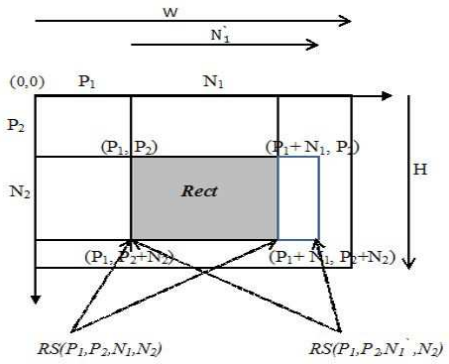


Fig. 3: Rectangle Sums sharing same height N_2. Two rectangle sums use the same Strip Subtraction or Strip Division for computation.

4.2 Horizontal and Vertical Strip Division

In order to achieve strip division in both horizontal and vertical directions for rectangle sum, the cumulative division operation is applied on the intensity values of every point in the image. We repeat the same process that we have discussed in Section 4.1. However, in this case, the basic arithmetic operation would be division rather than subtraction or sum operation. Let us consider, the gray value of a pixel (x,y) in an image be $g(x,y)$. Then cumulative division $CD(x,y)$ would be cumulatively divided value of all the gray values in the rectangle area from origin the $(0,0)$ to (x,y).

To calculate the image cumulative division pixel value for any location (x,y) the original image value at (x,y) is divided by the updated value (using cumulative division) of the pixel $(x-1,y)$ and it is also divided by original pixel value of all the pixels from $(x,0)$ to $(x,y-1)$. The algorithm of the cumulative division is given in Algorithm 2. Therefore, cumulative division can be defined as follows

$$CD(x,y) = ((x,y)/CD(x-1,y))/g(x,i) \tag{16}$$

where $i = 0,1,...,y-1$

Algorithm 2 Cumulative Division (CD)

```
Let, CD(0,0) = g(0,0) and d, c are two matrices having all zeros with size h × w.
for x = 0 to h − 1 do
    for y = 0 to w − 1 do
        b(x,y) = g(x,y)
    end for
end for
for x = 1 to h − 1 do
    ComputeCD(x,0) = g(x,0)/CD(x − 1,0)
end for
for y = 1 to w − 1 do
    for k = 0 to y − 1 do
        d(0,y) = b(0,k)/d(0,y)
    end for
    ComputeCD(0,y) = g(0,y)/d(0,y)
end for
for x = 1 to h − 1 do
    for y = 0 to w − 1 do
        for k = 0 to y − 1 do
            c(x,y) = b(x,k)/c(x,y)
        end for
        CD(x,y) = (g(x,y)/CD(x − 1,y))/c(x,y)
    end for
```

Horizontal Strip Division $HSD(P_1,P_2,N_2)$ in Figure 3 can be defined as the sum of pixels $X(m,n)$ for $0 \le m \le P_1 - 1$, $P_2 \le n \le P_2 + N_2 - 1$ with the fixed height N_2 and any width N_1. $HSD(P_1,P_2,N_2)$ can be computed by only one addition per pixel as follows using the Cumulative Division defined in Equation (16).

$$HSD\ (P_1,P_2,N_2) = \sum_{m=0}^{P_1-1}\sum_{n=P_2}^{P_2+N_2-1} X(m,n) = \cdots \tag{17}$$
$$\Sigma \quad \sum_{m=0}^{P_1-1}\sum_{n=0}^{P_2+N_2-1} X(m,n) - \sum_{m=0}^{P_1-1}\sum_{n=0}^{P_2-1} X(m,n) \cdots$$
$$CD \quad (P_1,P_2+N_2) - CD(P_1,P_2)$$

In order to calculate rectangle sum of *Rect* area $RS(P_1,P_2,N_1,N_2)$ and $RS(P_1,P_2,N_1',N_2)$, where $(N_1 \neq N_1')$ for $0 \le P_1$, $P_1 + N_1 \le W$; $P_1 + N_1' \le W$; $0 \le P_2$, $P_2 + N_2 \le H$, then we can write

$$RS \quad (P_1,P_2,N_1,N_2) = [CD(P_1+N_1,P_2+N_2) - \cdots$$
$$CD \quad (P_1+N_1,P_2)] - [CD(P_1,P_2+N_2) - \cdots$$
$$CD \quad (P_1,P_2)] = HSD(P_1+N_1,P_2,N_2) - \cdots \tag{18}$$
$$HSD\ (P_1,P_2,N_2)$$

$$RS \quad (P_1,P_2,N_1',N_2) = HSD(P_1+N_1',P_2,N_2) - \cdots$$
$$HSD\ (P_1,P_2,N_2) \tag{19}$$

Vertical Strip Division $VSD(P_1,P_2,N_1)$ can be defined as the sum of pixels $X(m,n)$ for $P_1 \le m \le P_1 + N_1 - 1$,

$0 \leq n \leq P_2 - 1$ with the any height N_2 and fixed width N_1. $VSD(P_1, P_2, N_1)$ can be computed by only one addition per pixel as follows using the Cumulative Division defined in Equation (16).

$$VSD\ (P_1,P_2,N_1) = \sum_{m=P_1}^{P_1+N_1-1}\sum_{n=0}^{P_2-1} X(m,n) = \cdots$$
$$\Sigma \sum_{m=0}^{P_1+N_1-1}\sum_{n=0}^{P_2-1} X(m,n) - \sum_{m=0}^{P_1-1}\sum_{n=0}^{P_2-1} X(m,n) \cdots \quad (20)$$
$$CD\ (P_1+N_1,P_2) - CD(P_1,P_2)$$

To calculate rectangle sum of *Rect* area $RS(P_1, P_2, N_1, N_2)$, where for $0 \leq P_2, P_2 + N_2 \leq H$; $0 \leq P_1$, $P_1 + N_1 \leq W$, we can write the expression as

$$RS\ (P_1,P_2,N_1,N_2) = [CD(P_1+N_1,P_2+N_2) - \cdots$$
$$CD\ (P_1,P_2+N_2)] - [CD(P_1+N_1,P_2) - \cdots$$
$$CD\ (P_1,P_2)] = VSD(P_1,P_2+N_2,N_1) - \cdots \quad (21)$$
$$VSD\ (P_1,P_2,N_1)$$

We can write $RS(P_1, P_2, N_1, N_2') = VSD(P_1, P_2 + N_2', N_1) - VSD(P_1, P_2, N_1)$, for sliding N_2 where $N_2' > N_2$ and obtain Haar Projection Values (HPVs) from the transformed image values using Haar basis function. Further, rectangle sum is determined using horizontal strip division and vertical strip division operations which are derived from the Equations (18) and (21).

4.3 Pattern Matching using Strip Subtraction and Strip Division

This section exploits pattern matching methodology on the use of strip subtraction and strip division operations characterized by orthogonal Haar Transform (OHT). To accomplish pattern matching with a pattern shown in Figure 4(b), the pattern is to be searched in the input image Figure 4(a). In order to initiate the process, the input image is divided into a number of candidate windows (also known as sliding window) as that of the size of the pattern. All the candidate windows need to be compared with the pattern to check the presence of the pattern in the image. At first, cumulative subtraction or cumulative division is applied on the pattern and also on the candidate windows. Then 2-D Haar representation is computed for both the pattern and candidate windows either using horizontal strip subtraction or horizontal strip division or vertical strip subtraction or vertical strip division. Each candidate window is now compared with the pattern at a time to find the *SSD (dissimilarity between pattern and selected candidate window)*. SSD_{min} is determined from all *SSD* scores to find the threshold (T) using $T = V \times SSD_{min} + N$, where V is the variable number, N is the number of pixels present in the pattern. Let us consider, during the experiment it has been determined at some instant that $d(\bar{X}_t, \bar{X}_w^j) \leq T$ (where both pattern and candidate windows are represented by vectors $\bar{X}_t$ and $\bar{X}_w^j$, respectively), then $\bar{X}_w^j$ (j^{th} candidate window) will be labelled as *'matched candidate window'*.

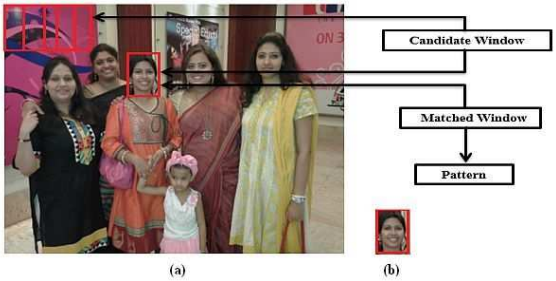


Fig. 4: 4(a) shows pattern, candidate windows and matched window, and 4(b) shows pattern.

5 Algorithm Complexity

This section presents time complexity analysis of strip subtraction and strip division for computing rectangle sum with orthogonal Haar transform (OHT). We initiate the analysis by assuming that N_2 is fixed. Then we can calculate $HSSub(x,y)$ and $HSD(x,y)$ without giving extra efforts. However, when the image size is given by $W \times H$, the complexity to calculate all $HSSub_{N_2}$ and HSD_{N_2} would be $O(W(H - N_2 + 1))$ for N_2. For example, if I is an image of size 4×4 and $N_2 = 2$, then complexity becomes $O(4(4 - 2 + 1)) = O(4 \times 3)$. It means when $W = 4$, each rectangle sum would be determined with respect to other three rectangle sum corresponds to three arithmetic subtraction operations. If H varies arbitrarily, then complexity becomes $O(WH(H-1)/2)$. For a pattern of $2^n \times 2^n$ size needs $(n+1)$ different heights $(h = 1, 2, .., 2^n)$. A pattern of $2^1 \times 2^1$ size, (1+1)=2 different heights are needed $(h = 1, 2)$, from (x,y) to $(x, y+h)$. Here, $h = 2$, point change with respect to different height would be $(x,y) \rightarrow (x,y+1) \rightarrow (x,y+2)$ and complexity would become $O([(n+1)(H+1) - 2^{n+1} + 1]W)$. So, for an image of size 4×4 and pattern of dimension 2×2, the complexity would become $O(7 \times 4)$. Because, from (x,y) to $(x+w, y+h)$, three subtractions are needed to obtain the rectangle sum; from $(x,y+1)$ to $(x+w, y+h)$, three subtractions are needed to obtain rectangle sum and finally from $(x,y+2)$ to $(x+w, y+h)$ = [(number of subtractions for (x,y) to $(x+w, y+h)$), (1 subtraction) and (number of subtractions for $(x,y+1)$ to $(x+w, y+h)$], three subtractions are needed to obtain the rectangle sum.

6 Evaluation

This section describes the performance of the proposed pattern matching techniques strip subtraction and strip division methods using rectangle sum and Orthogonal Haar Transform (OHT) and results are determined on two databases, such as a local database and MIT-CSAIL database [23] of objects and scenes. The local database is prepared by collecting most of the image objects from World Wide Web (WWW) and rest of them from a personal photo library. Some sample images from the local database are shown in Figure 5. MIT-CSAIL database contains indoor and outdoor objects captured in office and urban environments. However, a part of the database is prepared by collecting images from the web. The database contains thousands of static images and sequences with 2500 annotated frames and it provides annotations for more than 30 objects. The images are captured by webcam as well as by digital camera. Some sample images from MIT-CSAIL database are shown in Figure 6.

Fig. 5: Some sample images from local database of objects and scenes are shown.

As this has been reported in [10], [13], [15] that the Orthogonal Haar Transform (OHT) performs better than the full search or full search equivalent algorithms, therefore to compute Haar-like features using strip subtraction and strip division would lead to construct better pattern searching algorithms and the sums of pixels within a rectangle (rectangle sums) are the building blocks for Haar-like features.

Fig. 6: Some sample scene images from MIT-CSAIL database are shown.

Table 1: Image data sets, different sizes of corresponding scene images and different sizes of patterns for Local Database are specified.

Data Set	Image Size	Pattern / Template Size
S1	160×120	4×4
S2	160×120	8×8
S3	256×256	12×12
S4	256×256	16×16
S5	640×480	16×16
S6	1280×960	16×16

6.1 Experimental Results on Local Database

In order to evaluate the OHT using both strip subtraction and strip division methods for searching a pattern in the scene images, the efficacy of the proposed algorithms is tested with six different image data sets prepared from the local database. The six data sets containing images of different sizes and patterns of four different sizes, viz. 4×4, 8×8, 12×12 and 16×16 are used for experiments are shown in Table 1. In order to perform the test and measure the dissimilarity between reference pattern and matching window, sum-of-squared-differences (SSD) [14] metric is used. The dissimilarity score happens to be compared with a threshold (T) and it can be defined by $T = 1.2 \times SSD_{min} + N$, where N is the number of pixels in a pattern and SSD_{min} defines minimum of sum-of-squared-differences between the reference pattern and the best matching window [14].

To understand the nature of coefficients of transformed pattern for pattern searching using trade-off between pixel positions and intensity values in the pattern prior to obtain Haar Projection Values (HPVs) and after obtaining Haar Projection Values, three techniques such as cumulative addition, cumulative subtraction and cumulative division are used to determine range of values

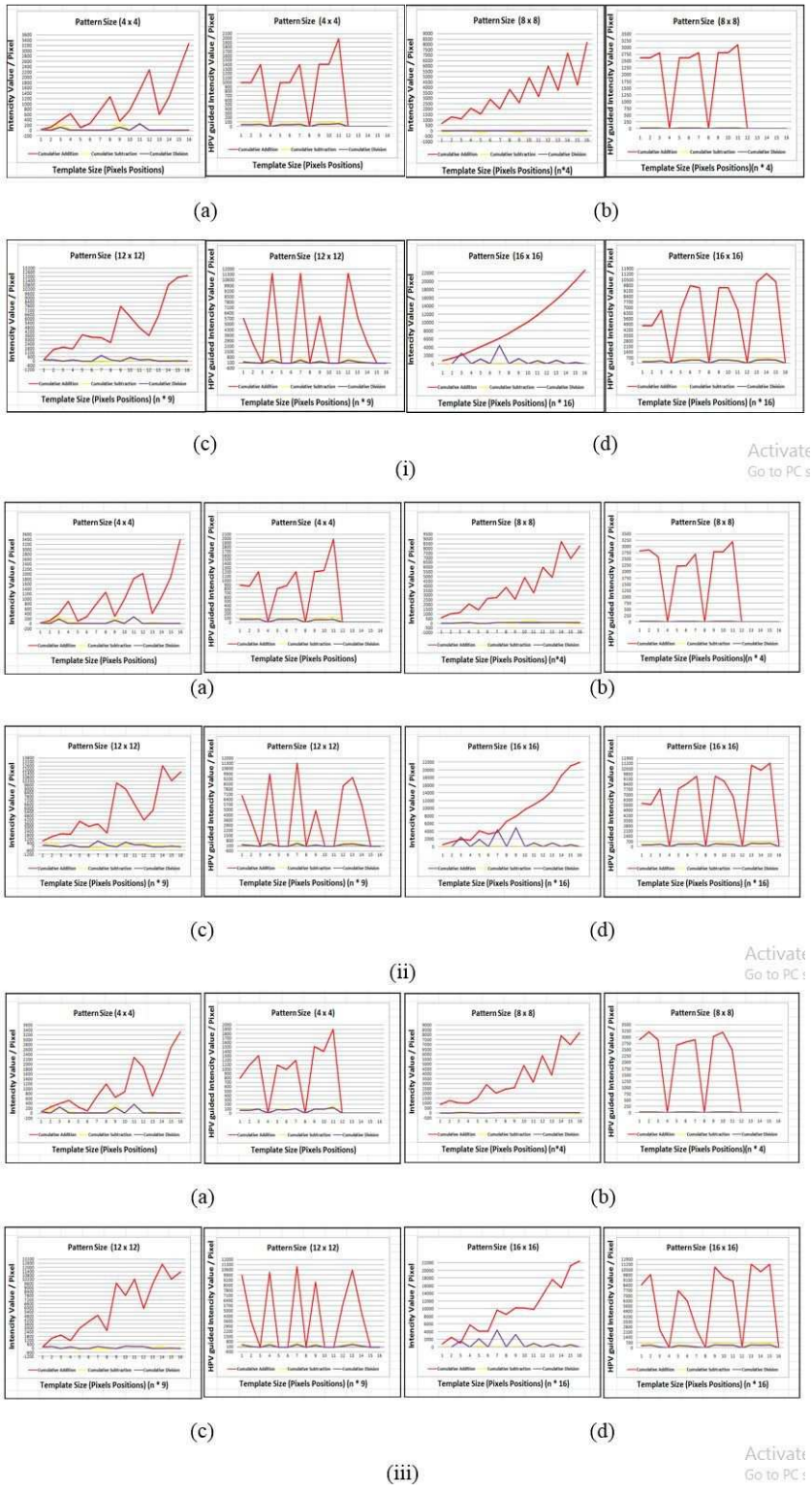
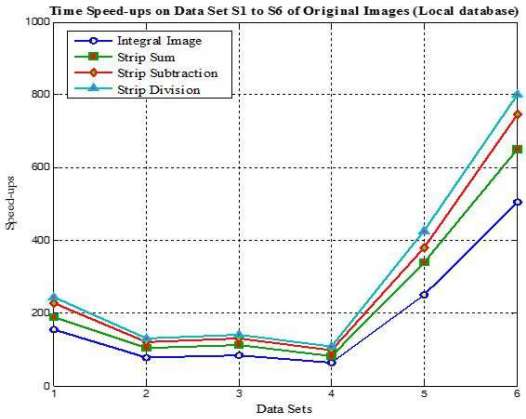
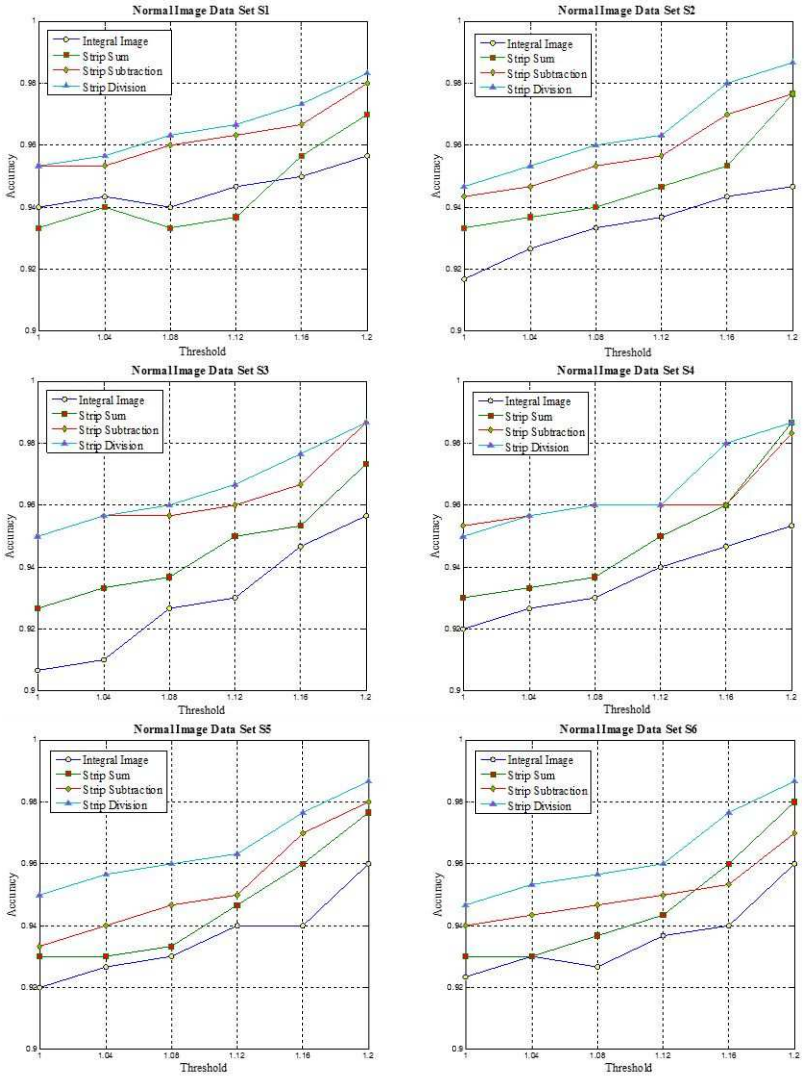


Fig. 7: Cumulative addition, cumulative subtraction and cumulative division algorithms are used to determine range of values on local database using different size of patterns ((a) 4×4,(b) 8×8, (c) 12×12 and (d) 16×16).



(a)



(b)

Fig. 8: (a) Time Speed-ups measurements of integral image, strip sum, strip subtraction and strip division on local datasets from *S1* through *S6* of normal images are shown. (b) The trade-offs between accuracy and threshold on local datasets ranging from *S1* through *S6* containing normal images, on which integral image, strip sum, strip subtraction and strip division are tested, are shown.

Table 2: TPR, FPR, TNR, FNR and pattern matching accuracy determined on datasets from S1 through S6 of local database of normal images at optimal threshold, are shown.

Dataset	Method	TPR	FPR	TNR	FNR	Accuracy
S1	Integral Image	0.978814	0.125	0.875	0.021186	0.956667
	Strip Sum	0.985401	0.192308	0.807692	0.014599	0.97
	Strip Subtraction	0.989209	0.136364	0.863636	0.010791	0.98
	Strip Division	0.989209	0.090909	0.909091	0.010791	0.983333
S2	Integral Image	0.979079	0.180328	0.819672	0.020921	0.946667
	Strip Sum	0.988806	0.125	0.875	0.011194	0.976667
	Strip Subtraction	0.985294	0.107143	0.892857	0.014706	0.976667
	Strip Division	0.992647	0.071429	0.928571	0.007353	0.986667
S3	Integral Image	0.967871	0.098039	0.901961	0.032129	0.956667
	Strip Sum	0.988679	0.142857	0.857143	0.011321	0.973333
	Strip Subtraction	0.992424	0.055556	0.944444	0.007576	0.986667
	Strip Division	0.992509	0.060606	0.939394	0.007491	0.986667
S4	Integral Image	0.982979	0.153846	0.846154	0.017021	0.953333
	Strip Sum	0.992126	0.043478	0.956522	0.007874	0.986667
	Strip Subtraction	0.992509	0.090909	0.909091	0.007491	0.983333
	Strip Division	0.992593	0.066667	0.933333	0.007407	0.986667
S5	Integral Image	0.983607	0.142857	0.857143	0.016393	0.96
	Strip Sum	0.98855	0.105263	0.894737	0.01145	0.976667
	Strip Subtraction	0.992565	0.129032	0.870968	0.007435	0.98
	Strip Division	0.992537	0.0625	0.9375	0.007463	0.986667
S6	Integral Image	0.98008	0.142857	0.857143	0.01992	0.96
	Strip Sum	0.992395	0.108108	0.891892	0.007605	0.98
	Strip Subtraction	0.984962	0.147059	0.852941	0.015038	0.97
	Strip Division	0.992647	0.071429	0.928571	0.007353	0.986667

Table 3: Different sizes of scene images and patterns from MIT-CSAIL database are used in datasets S1 through S6 in the experiment.

Data Set	Image Size	Pattern / Template Size
S1	160×120	16×16
S2	320×240	32×32
S3	640×480	64×64
S4	1280×960	128×128
S5	1280×960	64×64
S6	1280×960	32×32

on the local database using four different sizes of patterns (4×4, 8×8, 12×12 and 16×16) shown in Figure 7. It has been noticed that OHT calculation using strip subtraction (using cumulative subtraction) and strip division (using cumulative division) need very less amount of time for execution with respect to strip sum (using cumulative addition) and the intensity of each pixel (after applying cumulative subtraction or cumulative division for strip subtraction and strip division respectively), HPV guided intensity value of each pixel are very less in compare to generated intensity value of each pixel (after applying cumulative addition for strip sum), that means that values represented by strip subtraction and strip division would take very less amount of memory space. The results are shown with respect to pattern size of four different data sets. Similarly, we can show that strip subtraction and strip division reduces the time and memory requirement in comparison with strip sum for different size images or windows.

In Figure 7, we can see that strip sum using cumulative addition takes a higher value range for all the data sets, whereas strip subtraction using cumulative subtraction (CS) and strip division using cumulative division (CD) represent the image intensity value along with HPV of the image in very small range of value. Thus memory requirement using cumulative subtraction and cumulative division for strip subtraction and strip division respectively is very low in compare to cumulative addition for strip sum. Therefore, the rectangle sum of a particular rectangle region calculated by the cumulative sum for strip sum is much higher than the rectangle sum value calculated by two techniques (CS and CD).

Figure 8(a) shows that the speedup for the entire process of pattern matching using the existing techniques such as integral image and strip sum, and proposed techniques such as strip subtraction and strip division. We can understand from Figure 8(a) that pattern matching process using strip subtraction and strip division are

Table 4: TPR, FPR, TNR, FNR and pattern matching accuracy are determined on six datasets ranging from S1 to S6 are containing normal images prepared from MIT-CSAIL database, at optimal threshold, are shown.

Dataset	Method	TPR	FPR	TNR	FNR	Accuracy
S1	Integral Image	0.979592	0.181818	0.818182	0.020408	0.95
	Strip Sum	0.989011	0.185185	0.814815	0.010989	0.973333
	Strip Subtraction	0.992701	0.115385	0.884615	0.007299	0.983333
	Strip Division	0.99278	0.086957	0.913043	0.00722	0.986667
S2	Integral Image	0.979079	0.213115	0.786885	0.020921	0.94
	Strip Sum	0.988806	0.15625	0.84375	0.011194	0.973333
	Strip Subtraction	0.992593	0.1	0.9	0.007407	0.983333
	Strip Division	0.992647	0.107143	0.892857	0.007353	0.983333
S3	Integral Image	0.979253	0.220339	0.779661	0.020747	0.94
	Strip Sum	0.988417	0.121951	0.878049	0.011583	0.973333
	Strip Subtraction	0.992424	0.083333	0.916667	0.007576	0.983333
	Strip Division	0.992509	0.090909	0.909091	0.007491	0.983333
S4	Integral Image	0.978814	0.203125	0.796875	0.021186	0.94
	Strip Sum	0.988235	0.111111	0.888889	0.011765	0.973333
	Strip Subtraction	0.992509	0.090909	0.909091	0.007491	0.983333
	Strip Division	0.992593	0.1	0.9	0.007407	0.983333
S5	Integral Image	0.979424	0.175439	0.824561	0.020576	0.95
	Strip Sum	0.98855	0.131579	0.868421	0.01145	0.973333
	Strip Subtraction	0.992565	0.16129	0.83871	0.007435	0.976667
	Strip Division	0.992537	0.09375	0.90625	0.007463	0.983333
S6	Integral Image	0.98008	0.204082	0.795918	0.01992	0.95
	Strip Sum	0.988636	0.138889	0.861111	0.011364	0.973333
	Strip Subtraction	0.988679	0.142857	0.857143	0.011321	0.973333
	Strip Division	0.992481	0.088235	0.911765	0.007519	0.983333

Table 5: TPR, FPR, TNR, FNR and pattern matching accuracy are determined on six datasets ranging from *S*1 to *S*6 are containing noisy images prepared from MIT-CSAIL database, at optimal threshold, are shown.

Dataset	Method	TPR	FPR	TNR	FNR	Accuracy
S1	Integral Image	0.977778	0.186667	0.813333	0.022222	0.936667
	Strip Sum	0.988235	0.133333	0.866667	0.011765	0.97
	Strip Subtraction	0.992337	0.153846	0.846154	0.007663	0.973333
	Strip Division	0.992308	0.075	0.925	0.007692	0.983333
S2	Integral Image	0.979592	0.236364	0.763636	0.020408	0.94
	Strip Sum	0.988636	0.166667	0.833333	0.011364	0.97
	Strip Subtraction	0.992509	0.090909	0.909091	0.007491	0.983333
	Strip Division	0.992593	0.1	0.9	0.007407	0.983333
S3	Integral Image	0.979592	0.236364	0.763636	0.020408	0.94
	Strip Sum	0.988593	0.135135	0.864865	0.011407	0.973333
	Strip Subtraction	0.988593	0.108108	0.891892	0.011407	0.976667
	Strip Division	0.99262	0.068966	0.931034	0.00738	0.986667
S4	Integral Image	0.978723	0.2	0.8	0.021277	0.94
	Strip Sum	0.988593	0.108108	0.891892	0.011407	0.976667
	Strip Subtraction	0.99262	0.068966	0.931034	0.00738	0.986667
	Strip Division	0.992509	0.090909	0.909091	0.007491	0.983333
S5	Integral Image	0.979592	0.181818	0.818182	0.020408	0.95
	Strip Sum	0.992509	0.090909	0.909091	0.007491	0.983333
	Strip Subtraction	0.992647	0.178571	0.821429	0.007353	0.976667
	Strip Division	0.992565	0.096774	0.903226	0.007435	0.983333
S6	Integral Image	0.980159	0.208333	0.791667	0.019841	0.95
	Strip Sum	0.988764	0.151515	0.848485	0.011236	0.973333
	Strip Subtraction	0.988636	0.138889	0.861111	0.011364	0.973333
	Strip Division	0.992593	0.1	0.9	0.007407	0.983333

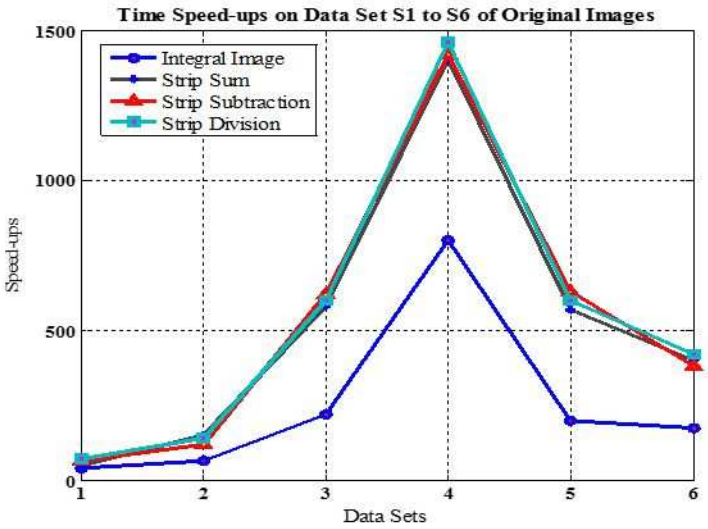


Fig. 9: Time Speed-ups measurements on six datasets S1 through S6 of original images are shown.

giving a competitively better speedup compare to strip sum and integral image techniques. In Figure 8(b), we can see at the threshold level (level 6) for accuracy measurement for all datasets, strip subtraction and strip division outperform the Integral Image. For datasets 4 and 6, strip sum has better accuracy than strip subtraction and for dataset 4, strip sum has the same accuracy as that of strip division. But for the other cases, both strip subtraction and strip division exhibit better performance than strip sum. In order to show the efficacy of the proposed techniques, viz. strip subtraction and strip division on local database, and compare them with two existing techniques integral image and strip sum, the values of essential parameters are determined such as True Positive Rate (TPR), False Positive Rate (FPR), True Negative Rate (TNR), False Negative Rate (FNR) and pattern matching accuracy. Table 2 shows the performance of strip subtraction and strip division techniques along with integral image and strip sum in terms of TPR, FPR, TNR, FNR and pattern matching accuracy. In can be seen from Table 2 that, in most of the cases, strip subtraction and strip division outperform integral image and strip sum methods except few ones while pattern matching accuracy is taken into consideration. The results are determined for the optimal threshold. Other non-optimal thresholds can be used to determine the parameter values. However, that would not be good results.

6.2 Experimental Results on MIT-CSAIL Database

In the next experiment, we have tested the performance of two proposed algorithms strip subtraction and strip division along with existing pattern matching algorithms such as integral image and strip sum on six datasets prepared from MIT-CSAIL database [23] and Table 3 show the different size of scene images and patterns. The experimental results determined on MIT-CSAIL database using strip subtraction and strip division methods are also compared with integral image [10] and strip sum [15] methods. Six datasets are prepared with 120 Images taken from MIT-CSAIL database. These six datasets are uniformly sampled and four different sizes of scene images, viz. 160×120, 320×240, 640×480 and 1280×960 are considered and each 30 images have the same resolution. Similarly, four different sizes of patterns (16×16, 32×32, 64×64 and 128×128) are used in the experiment. For each image, 10 different randomly selected patterns are considered. So each dataset will have 300 pattern image pairs. The proposed methods along with the existing methods are tested with three different image types: original images, noisy images, and blurred images. Datasets *S*5 and *S*6 are used for scrutinizing the consequences of pattern size in pattern matching process with same scene image size varying pattern size.

For each test on a particular image type, *SSD* is used to find degree of similarity when the threshold is set as

$$T = 1.1 \times SSD_{min} + N, \tag{22}$$

where N is the number of pixels present in pattern image and SSD_{min} represents the minimum of *SSD* in between pattern and best matching candidate window.

Following three sub-sections will summarize the experimental results determined on original, noisy and blur images prepared from MIT-CSAIL database. As only the original images are given in the database, however for

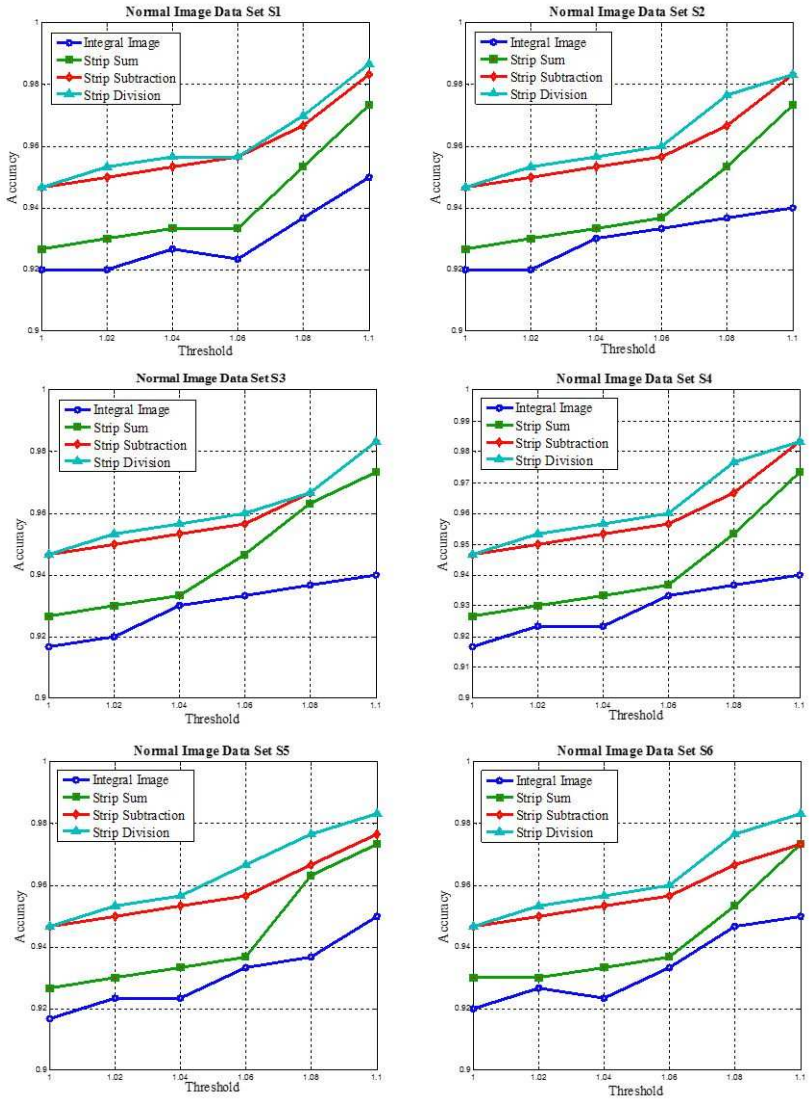


Fig. 10: The trade-offs between accuracy and threshold determined on six datasets ranging from *S*1 through *S*6 containing normal images prepared from MIT-CSAIL database on which strip subtraction, strip division, integral image and strip sum algorithms are tested, are shown.

Fig. 11: An original image with noisy images at four different levels are shown.

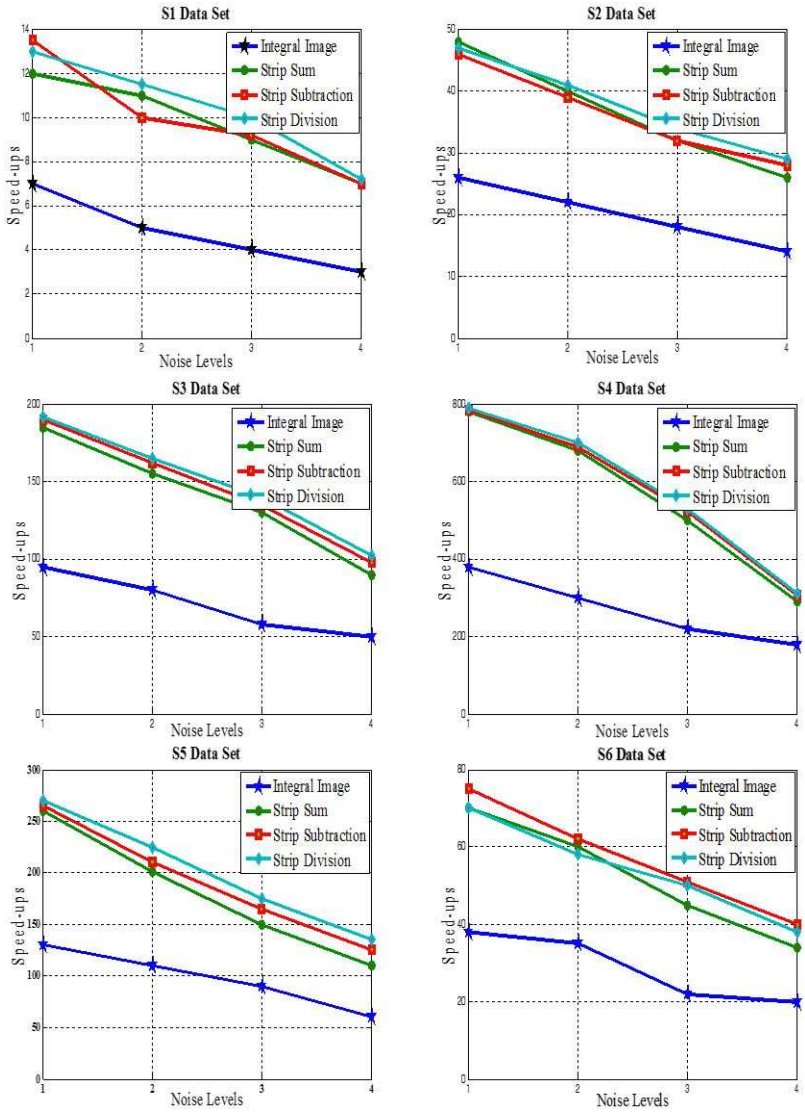


Fig. 12: Speed-ups in execution time on six datasets ranging from $S1$ to $S6$ with four different noise levels.

the experiment, noisy and blur images are being generated and performance parameters are measured.

6.2.1 Experimental results on original images

As per the distribution of database into six datasets described in Table 3, the experiment with the proposed techniques strip subtraction and strip division along with the existing techniques strip sum and integral image is conducted on original images in order to understand and exhibit the performance in terms of speed against the datasets, shown in Figure 9. The time or operation speedup of algorithm X over algorithm Y is given by

$$= \frac{execution time or number of operations required by Y}{execution time or number of operations required by X} \quad (23)$$

From Figure 9 we can see that both strip subtraction and strip division are found to be much faster than integral image and almost overlapping speed-ups compare to strip sum method for a small size of the scene image. However, pattern matching for a large size of scene image with a large size of the pattern (dataset $S4$) or varying pattern size (datasets $S5$ and $S6$), both strip subtraction and strip division methods perform better having improved speed-ups compare to strip sum.

While the proposed pattern matching algorithms, viz. strip subtraction and strip division are tested on six

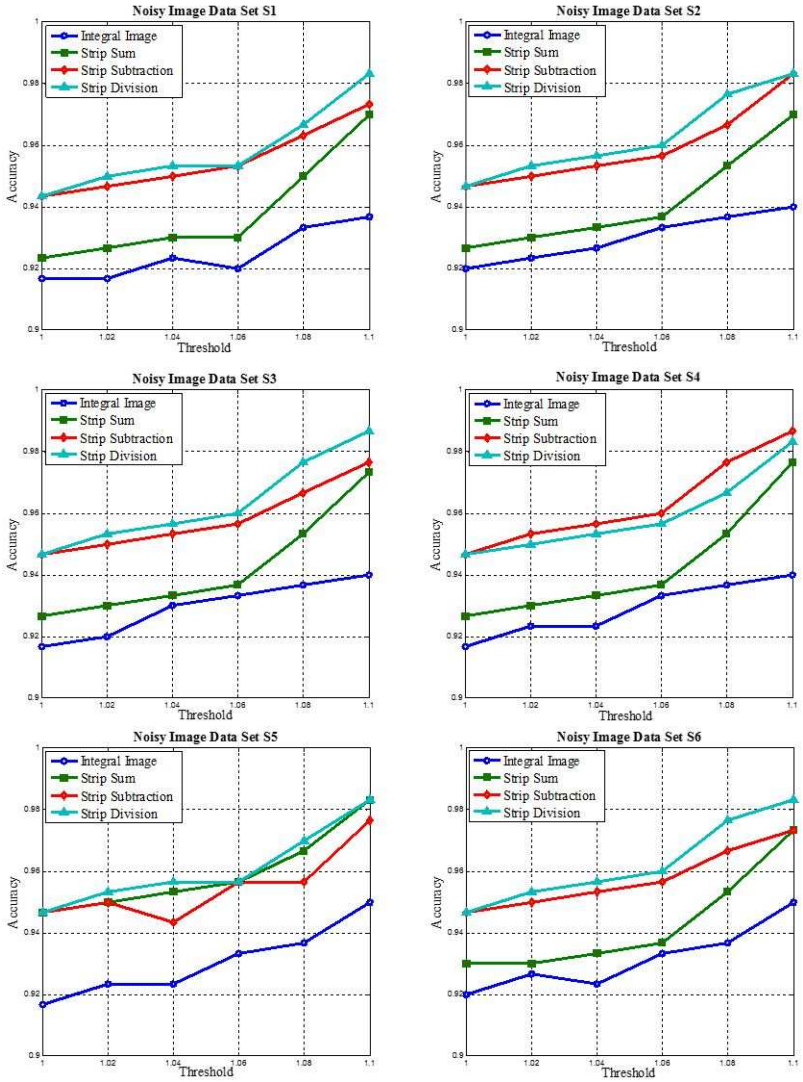


Fig. 13: The trade-offs between accuracy and threshold determined on six datasets ranging from *S*1 through *S*6 containing noisy images prepared from MIT-CSAIL database on which strip subtraction, strip division, integral image and strip sum algorithms are tested, are shown.

different datasets of normal images prepared from MIT-CSAIL database, a comparison is also presented with two existing algorithms on the same datasets to exhibit the efficacy of the proposed algorithms. During the experiment, it has been observed that both strip subtraction and strip division outperform integral image as well as strip sum algorithms on first five datasets in terms of pattern matching accuracy. However, for dataset *S*6, both strip subtraction and strip division along with strip sum are found competitive to each other. Table 4 shows TPR, FPR, TNR, FNR and accuracy which are determined on six datasets using the proposed and existing algorithms. In case of the integral image, accuracy slows down when the algorithm is evaluated on six datasets of normal images.

The trade-off between accuracy and threshold levels determined on six datasets of normal images are shown in Figure 10 where accuracy is determined for six different thresholds on each dataset and curves are plotted for the proposed as well as for the existing algorithms. However, Table 4 shows the accuracy which is given for optimal threshold only when threshold varies between 1.0 and 1.1.

6.2.2 Experimental results on noisy images

To assess the performance of the algorithms on noisy images, four low-to-high levels (i.e., N_1, N_2, N_3 and N_4) of iid zero-mean Gaussian noise are added to every image of the six datasets in featured in Table 3. These 4-level of

Fig. 14: Four levels of blurred images correspond to an original image are depicted.

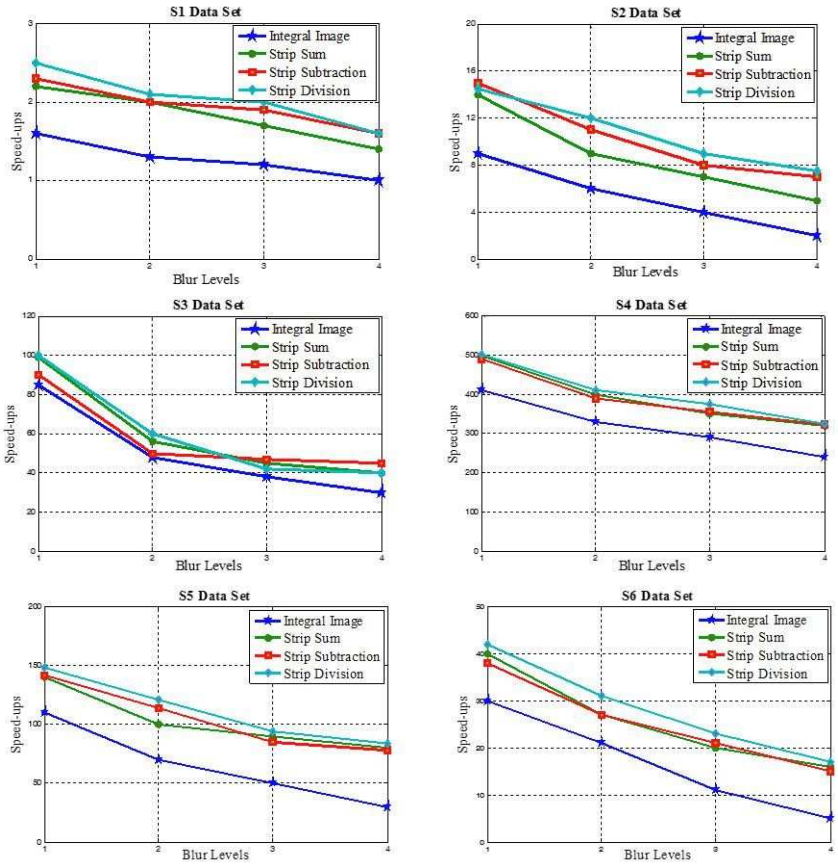


Fig. 15: Speed-ups in execution time on datasets $S1$ through $S6$ having four different blur levels.

Gaussian noises N_1, N_2, N_3 and N_4 range from low to high noise with deviations 100, 200, 400 and 800 respectively. With respect to one original image, how it changes with applying of four different noises are depicted in Figure 11.

The time speed-ups of the proposed algorithms over existing algorithms in different size of patterns and scene images with four different noise levels are depicted in Figure 12. It has been seen that strip subtraction and strip division are found very competitive to strip sum for datasets $S1$, $S2$ and $S3$, and faster than the integral image for all datasets. It can also be observed that strip subtraction and strip division perform better for large scene image (i.e., $S4$, $S5$ and $S6$). Moreover, when pattern small in size is compared to scene image, the proposed two algorithms are at their very best for images with increasing noise levels. So, the speed-ups are not only depending on different noise levels, but also on the different size of patterns and scene images.

Noisy images are generated by adding four different noise levels to normal images contained in MIT-CSAIL database and then the performance of the proposed algorithms are exhibited by determining TPR, FPR, TNR, FNR and accuracy on six datasets. The accuracy versus threshold curves determined noisy images are shown in Figure 13 and the experimental results are depicted in Table 5. It is being observed that, when algorithms are tested on dataset $S1$, strip subtraction has shown better

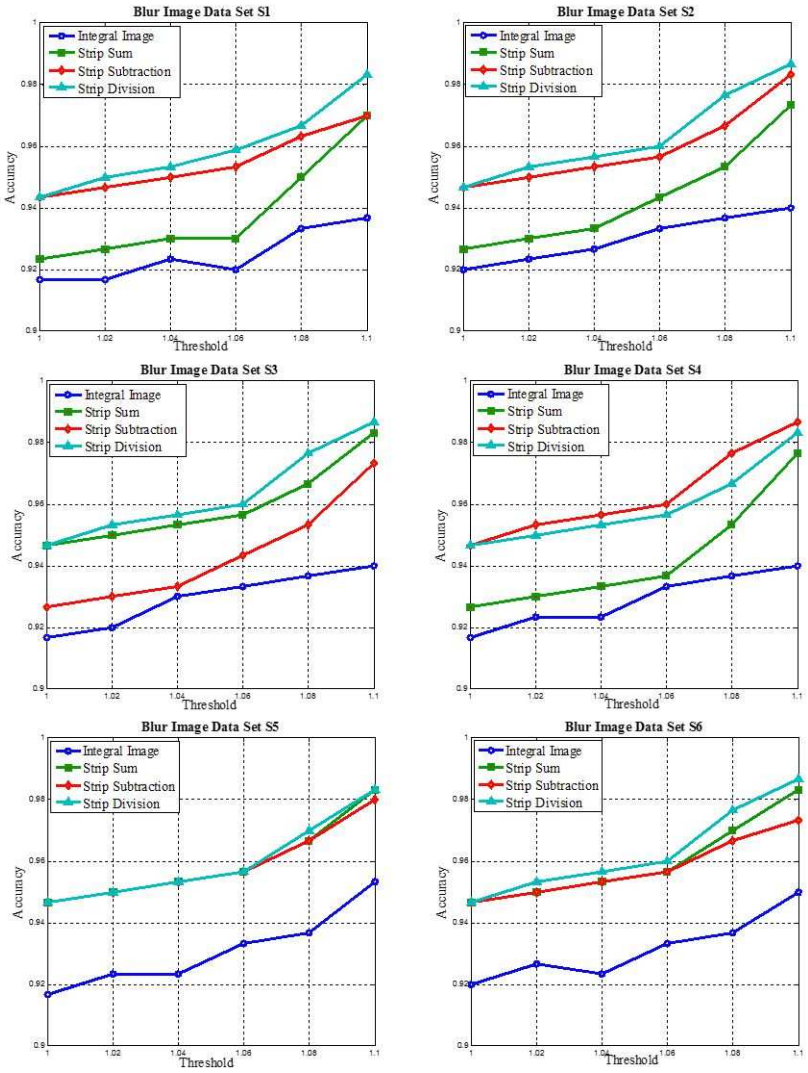


Fig. 16: The trade-offs between accuracy and threshold determined on six datasets ranging from *S*1 through *S*6 containing blur images prepared from MIT-CSAIL database on which strip subtraction, strip division, integral image and strip sum algorithms are tested, are shown.

TPR compared to strip sum, integral image and strip division methods, whereas strip division outperforms other three methods when accuracy is measured. On dataset *S*2, both strip subtraction and strip division have shown better TPR and accuracy compared to strip sum and integral image, however, both strip subtraction and strip division have the same accuracy. On dataset *S*3, strip division has shown better TPR and accuracy than the other methods, whereas both strip sum and strip subtraction have the same TPR and strip subtraction is found with better accuracy than strip sum and integral image. Both strip subtraction and strip division have determined better TPR and accuracy on dataset *S*4 compared to strip sum and integral image, however, strip subtraction exhibits better accuracy as well as better TPR than strip division. On dataset *S*5, both strip subtraction and strip division show better TPR than strip sum and integral image, whereas strip division and strip sum have determined higher accuracy compared to strip subtraction and integral image which are having the same accuracy. When algorithms are tested on dataset *S*6, strip sum has found lesser TPR than strip division and better TPR than strip subtraction and integral image, whereas strip division returns best accuracy compare to other methods and strip sum has the same accuracy as that of strip subtraction.

Table 6: TPR, FPR, TNR, FNR and pattern matching accuracy are determined on six datasets ranging from $S1$ to $S6$ are containing blur images prepared from MIT-CSAIL database, at optimal threshold, are shown.

Dataset	Method	TPR	FPR	TNR	FNR	Accuracy
S1	Integral Image	0.978261	0.2	0.8	0.021739	0.936667
	Strip Sum	0.988372	0.142857	0.857143	0.011628	0.97
	Strip Subtraction	0.992308	0.15	0.85	0.007692	0.973333
	Strip Division	0.992366	0.078947	0.921053	0.007634	0.983333
S2	Integral Image	0.979675	0.240741	0.759259	0.020325	0.94
	Strip Sum	0.98893	0.172414	0.827586	0.01107	0.973333
	Strip Subtraction	0.99262	0.103448	0.896552	0.00738	0.983333
	Strip Division	0.992593	0.066667	0.933333	0.007407	0.986667
S3	Integral Image	0.979675	0.240741	0.759259	0.020325	0.94
	Strip Sum	0.99262	0.103448	0.896552	0.00738	0.983333
	Strip Subtraction	0.98893	0.172414	0.827586	0.01107	0.973333
	Strip Division	0.992647	0.071429	0.928571	0.007353	0.986667
S4	Integral Image	0.979167	0.216667	0.783333	0.020833	0.94
	Strip Sum	0.988593	0.108108	0.891892	0.011407	0.976667
	Strip Subtraction	0.992481	0.058824	0.941176	0.007519	0.986667
	Strip Division	0.99262	0.103448	0.896552	0.00738	0.983333
S5	Integral Image	0.980315	0.195652	0.804348	0.019685	0.953333
	Strip Sum	0.992565	0.096774	0.903226	0.007435	0.983333
	Strip Subtraction	0.99262	0.137931	0.862069	0.00738	0.98
	Strip Division	0.992593	0.1	0.9	0.007407	0.983333
S6	Integral Image	0.980392	0.222222	0.777778	0.019608	0.95
	Strip Sum	0.992593	0.1	0.9	0.007407	0.983333
	Strip Subtraction	0.988764	0.151515	0.848485	0.011236	0.973333
	Strip Division	0.992647	0.071429	0.928571	0.007353	0.986667

6.2.3 Experimental results on blurred images

Similar to normal and noisy images, the proposed, as well as existing algorithms, are also evaluated on blurred images which are used four different levels (i.e., B_1, B_2, B_3 and B_4) of Gaussian low-pass filters to blur each scene image of the datasets featured in Table 3. The four blur levels correspond to Gaussian low-pass filters with standard deviations 0.9, 1.6, 2.3 and 3.0 respectively. With respect to one original image, how it changes or distorts with the effects of four different blur levels are depicted in Figure 14.

The time speed-ups curves of the proposed algorithms over existing algorithms with different size of patterns and scene images are shown in Figure 15. It is being seen that both strip subtraction and strip division are found very competitive when they are compared to strip sum and also found faster than the integral image in all cases. When the algorithms are tested on datasets $S1$, $S2$, $S3$ and $S5$, both strip subtraction and strip division show better achievable speed-ups for scene image with four different blur effects. As we can see that for the combination of large scene image and decreasing pattern size (datasets $S4$ & $S6$), strip subtraction and strip sum are found very much competitive to each other, however, strip division outperforms strip sum when speed-ups is measured.

The performance of the proposed algorithms strip subtraction and strip division along with the existing algorithms strip sum and integral image are also tested on images with different blur levels contained in six datasets. The experimental results and accuracy versus threshold curves are shown in Table 6 and in Figure 16 respectively. The pattern matching accuracies are determined against heuristically computed threshold having a range between 1.0 and 1.1 with interval length 0.02. However, the accuracies for all six datasets containing blur images are given in Table 6, are determined at an optimal threshold. In this experiment, different performance parameters such as TPR, FPR, TNR, FNR and accuracy, are determined on six datasets containing blur images prepared from MIT-CSAIL database. When the evaluation is performed on dataset $S1$, both strip subtraction and strip division have shown better TPR and accuracy compared to strip sum and integral image. When the experiment is conducted on dataset $S2$, strip subtraction exhibits better TPR than strip sum, integral image and strip division, however, on the other hand, strip division shows higher accuracy than other three methods. On datasets $S3$ and $S6$, strip sum exhibits lesser TPR and accuracy when it is compared to strip division, though it shows better TPR and accuracy with respect to strip subtraction and integral image. On dataset $S4$, strip division determines better TPR when it is compared to strip subtraction, strip sum and integral image, however, strip subtraction has determined better accuracy than other three pattern matching methods. When the experiment is performed on dataset $S5$, strip subtraction has shown better TPR compared to other methods, however, strip division and

strip sum exhibit better accuracy than strip subtraction and integral image while they possess the same accuracy.

7 Conclusion

This paper has reported some pattern matching techniques, viz. horizontal strip subtraction, horizontal strip division, vertical strip subtraction and vertical strip division methods which compute the sum of pixels available within a rectangle called rectangle sum. In order to obtain the Haar like features, rectangle sum is always found to be very important property. We have shown that horizontal strip subtraction, horizontal strip division, vertical strip subtraction and vertical strip division methods can be used to compute orthogonal Haar transform (OHT) in a very comprehensive way and then OHT is used for equivalent pattern matching where image cumulative subtraction and image cumulative division methods are representing input image in small values compared to values calculated by cumulative sum in image integral method. Also, the rectangle sum of a particular rectangle region calculated by the cumulative sum for strip sum is much higher than the rectangle sum value calculated by two techniques cumulative subtraction for horizontal strip subtraction, vertical strip subtraction and cumulative division for horizontal strip division and vertical strip division. The proposed techniques are compared with some existing techniques in determining the results on a local database and MIT-CSAIL databases. The results exhibit that the proposed techniques outperform the existing methods except for a few cases where similar effects have been recorded.

Acknowledgement

The authors are grateful to the anonymous referee for a careful checking of the details and for helpful comments that improved this manuscript.

References

[1] K. Ahuja and P. Tuli, Object recognition by template matching using correlations and phase angle method, International Journal of Advanced Research in Computer and Communication Engineering **2(3)**, 1768-1773 (2013).
[2] M. Alkhansari, A fast globally optimal algorithm for template matching using low-resolution pruning, IEEE Transactions on Image Processing **10(4)**, 526-533 (2001).
[3] G. Ben-Artz, H. Hel-Or and Y. Hel-Or, The gray-code filter kernels, IEEE Transactions on Pattern Analysis and Machine Intelligence **29(3)**, 382-393 (2007).
[4] Y. M. Fouda, A robust template matching algorithm based on reducing dimensions, Journal of Signal and Information Processing **6**, 109-122 (2015).
[5] Y. Hel-Or and H. Hel-Or, Real time pattern matching using projection kernels, IEEE Transactions on Pattern Analysis and Machine Intelligence **27(9)**, 1430-1445 (2005).
[6] M. I. Khalil, Quick techniques for template-matching-based cross-correlation, World Applied Sciences Journal **33(3)**, 430-436 (2015).
[7] S. Korman, D. Reichman, G. Tsur and S. Avidan, FAsT match: Fast affine template matching, In: Computer Vision and Pattern Recognition, Portland, (2013).
[8] C. Lampert, M. Blaschko and T. Hofmann, Beyond sliding windows: Object localization by efficient subwindow, In: Computer Vision and Pattern Recognition, (2008).
[9] J. Lewis, Fast template matching, In: Vision Interface, (1995).
[10] Y. Li, H. Li and Z. Cai, Fats orthogonal Haar transform pattern matching via image square sum, IEEE Transactions on Pattern Analysis and Machine Intelligence **36(9)**, 1748-1760 (2014).
[11] T. Mahalakshmi, R. Muthaiah and P. Swaminathan, An overview of template matching technique in image processing, Research Journal of Applied Sciences, Engineering and Technology **4(24)**, 5469-5473 (2012).
[12] S. Mattoccia, F. Tombari and L. D. Stefano, Fast full-search equivalent template matching by enhanced bounded correlation, IEEE Transactions on Image Processing **17(4)**, 528-538 (2008).
[13] W. Ouyang and W. K. Cham, Fast algorithm for Walsh Hadamard transform on sliding windows, IEEE Transactions on Pattern Analysis and Machine Intelligence **32(1)**, 165-171 (2010).
[14] W. Ouyang, F. Tombari, S. Mattoccia, L. D. Stefano and W. K. Cham, Performance evaluation of full search equivalent pattern matching algorithms, IEEE Transactions on Pattern Analysis and Machine Intelligence **34(1)**, 127-143 (2012).
[15] W. Ouyang, R. Zhang and W. K. Cham, Fast pattern matching using orthogonal Haar transform, In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2010).
[16] W. Pan, S. Wei and S. Lai, Efficient NCC-based image matching in Walsh-Hadamard domain, In: 10th European Conference on Computer Vision, (2008).
[17] O. Pele and M. Werman, Robust real-time pattern matching using Bayesian sequential hypothesis testing, IEEE Transactions on Pattern Analysis and Machine Intelligence **30(8)**, 1427-1443 (2008).
[18] L. D. Stefano and S. Mattoccia, Fast template matching using bounded partial correlation, Journal of Machine Vision and Applications **13**, 213-221 (2003).
[19] F. Tombari, S. Mattoccia and L. D. Stefano, Full search-equivalent pattern matching with incremental dissimilarity approximations, IEEE Transactions on Pattern Analysis and Machine Intelligence **31(1)**, 129-141 (2009).
[20] P. Viola and M. Jones, Rapid object detection using a boosted cascade of simple features, In: Computer Vision and Pattern Recognition, (2001).
[21] S. D. Wei and S. H. Lai, Fast template matching based on normalized cross correlation with adaptive multilevel winner update, IEEE Transactions on Image Processing **17(11)**, 2227-2235 (2008).
[22] T. Wu and A. Toet, Speed-up template matching through integral image based weak classifiers, Journal of Pattern Recognition Research **9(1)**, 1-12 (2014).

[23] A. Torralba, K. P. Murphy and W. T. Freeman, Sharing features: efficient boosting procedures for multiclass object detection, In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 762-769 (2004).

Deep Suman Dev received his B.E. (Information Technology) degree from Burdwan University, India in 2004, M.Tech. (Information Technology) degree from Indian Institute of Engineering, Science and Technology, Shibpur (IIEST), India formerly known as Bengal Engineering and Science University, Shibpur (BESU), India in 2007 and pursuing Ph.D. in Computer Science and Engineering from National Institute of Technology Durgapur, India. He is an Assistant Professor in the Department of Information Technology at Neotia Institute of Technology, Management and Science, India and he has been a faculty member since July, 2005. He has published 4 papers in refereed international journals and conferences. His research interest includes Computer Vision Application, Biometrics and Pattern Classification.

Dakshina Ranjan Kisku is currently an Assistant Professor in the Department of Computer Science and Engineering at National Institute of Technology Durgapur, India. Prior to joining this institute, he had been working as a faculty member at two reputed engineering colleges in India. He received his B.Eng., M.Eng. and Ph.D. (Engineering) degrees in Computer Science and Engineering from Jadavpur University, India. In 2006, he visited Computer Vision Laboratory at the University of Sassari, Italy and worked on biometrics recognition problems. He also visited IIT Kanpur in 2005 and worked on a sponsored project funded by MCIT, Government of India. He is a recipient of Sir Visvesvaraya Young Faculty Research Fellowship of Government of India in 2016, IEI Young Engineers Award of Institute of Engineers (India) in 2012, Ministry of Education, Universities and Research (MIUR) Research Fellowship of Government of Italy in 2006, International Travel Fellowship for Young Scientists of SERB India in 2010, IEEE/IAPR Student Endorsement Scholarship of IEEE in 2009 and IEEE Travel Award of IEEE Kolkata Section in 2010. He is a Senior Member of IEEE, and Member of ACM and SPIE. He has published more than 15 papers in refereed international journals, more than 50 papers in international conferences and more than 5 book chapters in edited books. He also co-authored and edited 3 books on biometrics and homeland security. His research interest includes biometrics, pattern classification, computer vision, image processing and digital forensics.