Abstract

Recently, image processing has been widely used in the medical field. This field helps to increase the accuracy of medical diagnosis, which helps in the early detection of diseases and increases the accuracy of prescribing treatment. Where it is noted that most medical devices such as sonar that are used to take pictures of body parts may give blurred images, which leads to the need for digital image processing techniques to increase the clarity of these images, which gives an accurate description of the examined part of the body. Speed is also one of the most important criteria for measuring the quality of any system, especially in the critical matters, including the medical field, where speed and accuracy of diagnosis are very important for immediate decision-making. To ensure the best speed and the most accurate result, it is better to distribute this process and use more than one processing unit for digital image processing. In this paper, a system was presented that detects the edges of medical images by relying on edge detection processing technology and taking advantage of the field of distributed systems to obtain results at high speed and accuracy. Matlab 2015 environment was used to simulate the system, and the results showed high accuracy in edge detection and high speed to obtain the results.
Keywords

Image Processing, Edge Detection, Distributed System, Color Space.

Introduction

Recently, there is an increasing flow of data, especially visual data. This increases the need to process large data sets quickly. In real-time image processing, useful information is always fed to the system instantly and needs to be processed in real-time as well. In the field of image processing, it is certainly a big challenge to design an efficient system to give a fast response (Wu, He, & Hintz, 2002). As automatic image segmentation as well as edge detection techniques are becoming more common and commonly used in many applications such as traffic light detection in ADAS systems, medical image diagnostic machine vision systems, and medical image diagnostic machine vision systems, etc. In general, the information on the object at edges or boundaries is available with high frequency noise or the artifact in boundaries because of incorrect image acquisition. Thus, it is very difficult to process or interpret such kinds of images (Janardhan, Ramanaiah, & Babulu, 2018). The most valuable impact from implementing content-based image retrieval, accessing all suites for images by describing consistent, scalable, human independent advantage based content. Because satisfying accuracy and retrieval can only be achieved with a high level of understanding of image features, a large set of features must be extracted, combined, and evaluated at different levels content abstraction process (Gueld et al., 2003).

Another role of this paper is to use the Distributed Prewitt Edge Detection Algorithm (DPEDA) to divide or detect object boundaries into more accurate. Edges basically contain an important message for the object. Over the decade, various solutions have been validated by various researchers to solve the problems of noise at image edges. The popular approach to all of these methods is nearly the same. This is the first noisy image to be filtered or smoothed using an appropriate filtering method. The second is to use either first order or second derivative methods to find the information of the local maximum.

Distributed computing is the practice of running computations on devices other than the one which initiated them. A distributed system is a logical abstraction that allows a single physical system to be viewed as if it were several physical systems. In distributed systems, critical concerns are efficient division and scheduling of tasks among the available processors so that the total processing time of the dispatched tasks is minimal. They have many benefits over traditional computer systems that use central processing units. Generally speaking, most distributed computer systems use a collection of computers to work together to solve complex problems that can't be solved by anyone single computer.
As technology progresses, there is an increased need for large amounts of resources that cannot be provided by a single system alone. This allows for a distributed computing system to provide one or more services like analyzing data, collecting information, and running software applications in areas where central processing units are not able to operate efficiently or at all without help from other computers around the world. Some computing systems are designed to make use of the resources provided by many computers working together in order to perform calculations or compute data. Distributed computing systems often consist of interconnected networks of machines with remote servers, which process information and then transmit it to other servers around the world.

The Prewitt edge detector is a simple and efficient edge detection algorithm that can be used in many applications including document image processing, video editing, and video surveillance. It is based on the gradient information from the image and a threshold on intensity. It is also known as a first-order differential operator, edge detector, or Laplace operator. The original system was designed for black-and-white images, but it can be corrected for color images. This edge detection algorithm is probabilistic and therefore its raw output has to be processed before it can be used as a decision-making tool. Prewitt edge detection system is very fast, requiring only one pass over the image, and does not need to compute the gradients of image intensity to do so. Prewitt algorithm will detect all edges, but some false positives may occur.

The Prewitt edge detection system was also used to compare two images and calculate the magnitude of their difference. Prewitt edge detection system detects changes in brightness at a pixel level. It isolates edges at which brightness changes abruptly from its surroundings using an algorithm that calculates the gradient magnitude. The Prewitt edge technique can be applied to any type of image and it’s widely used for recognizing shapes in photos, separating overlapping objects, detecting low-contrast edges between colors, or for other purposes involving shape analysis.

Rest of this paper organized as follows. Section 2, we examine related work. The edge detection and the prewitt operator description in Section 3. Then Section 4 introduces color spaces. suggest algorithm and result description in Section 5.

Related Work

In (Wu et al., 2002), An experimental study was conducted on the time performance of processing a large size of image data on a network of workstations. For the first time, such an experimental investigation has been conducted in the field of DLT. As a case study, it
uses edge detection using the Sobel operator as the application to prove the performance of the strategy suggested by DLT. Following the modeling of the system built, we contrast two different hashing strategies and scheduling strategies: first hash and scheduling strategy that uses PSSD theory and second is the traditional equal hashing strategy EQS. By looking at the empirical results as well as looking at the performance analysis using a different number of workstations, and image sizes, we note that time performance with PSSD is far superior to that gained using EQS.

According to (Janardhan et al., 2018), An improved Distributed Edge Detection (IDCEDA) is introduced to more precisely segment or detect and define object boundaries and is manufactured in an ISE environment the final design was developed by TSMC 0.18um technology. This design gave more perfect results with a minimum of no. of hardware resources compared to current approaches in terms of fewer hardware resources for implantation and accuracy. The algorithm performs better compared to existing methods in terms of hardware resource usage and sharp image limits.

In (Kazanskiy & Popov, 2012), they introduced the concept of image data distribution according to this concept that dynamically distributes the load of distributed of image processing to use a decentralized balancing algorithm. It also allows the analysis of distributed images on a native algorithm for dynamic load equilibrium of processors in point processing (PO) or local slip processing (LNO and RNO). The necessary level of fault tolerance is achieved in the storage of distributed image fragments. For most operations, analysis allows the user to create a new image without transmitting data between computers when doing distributed image processing.

In (Li et al., 2016), proposed a semantic dictionary to describe images at a semantic level. The semantic dictionary describes the probability distribution between semantic concepts and visual manifestations, the learning procedure for the semantic dictionary is formulated in a minimization optimization problems. The above-mentioned optimization solution was adopted by the mixed criteria organization to find out the concept of membership distribution of a visual appearance. In addition, a semantic expansion technique is introduced, to improve the generalization strength of semantic description, where a concept transfer matrix is learned to estimate the implicit association between concepts. The distributed framework was created on a basis of a semantic dictionary to quicken image understanding at scale. Semantic dictionary validation is done in image annotation tasks and large-scale semantic image search.
In (Zhang, Liu, Luo, & Lang, 2010), introduced a method for applying CBIR to large image datasets and the DIRS implementation system. A Distributed Image Retrieval System is created, images are retrieved by content-based manner, after which the retrieval process between the large images data storage is accelerated by using the MapReduce distributed computing model. Furthermore, the ability to operate in a heterogeneous environment, fault tolerance, and also the scalability are supported in the system that we created. Experiments are performed to verify performance improvement while using the MapReduce. Results show image storage and retrieval by using MapReduce outperforms in a highly centralized manner when a total number of the images is large.

According to (Yang, Jiang, & Song, 2018), they present a method for distributed image retrieval designed for a cloud computing based on the multi camera in a smart cities. By combining data encryption, data retrieval technology, and cloud storage technology, we the management of multiple camera resources and achieve effective integration. Cloud computing network data it will be released faster, which provides users with appropriate storage.

In (Sozykin & Epanchintsev, 2015), introduces the MapReduce Image Processing Framework, which provides the ability to use distributed computing for image processing. The MIPr is based on MapReduce and its open source application, Apache Hadoop. MIPr provides several forms of the image representations in Hadoop internal style and I/O tools for integrating the image processing into the Hadoop data performance. MIPr contains MapReduce implementations of common image processing algorithms, that can be used to process distributed images without any software development. MIPr framework greatly simplifies the image processing in a distributed Hadoop environment.

**Edge Detection**

In the area of digital image processing many edge detection techniques are used, each technique has its features and determinants. Edge detection techniques simply can be explained as using filters that pass on all the image elements, mathematically, edge detection apply a convolution process between the image elements and the filters to obtained the best results of edge detection (Ahmed Saadi Abdullah, Ali, & Waleed, 2019).

It is used in the edge detection process for many 2D filters. The edges are assumed to be pixels with a strong gradient in the edge detection algorithm. At edge pixels, a fast rate of change in pixel intensity in one direction is indicated by the angle of the gradient vector. An edge pixel is defined by two key characteristics: edge strength, which is equal to the
maximum of the gradient, and edge direction, which really is equal to the angle of the gradient. (Shrivakshan & Chandrasekar, 2012).

There are a plethora of edge detection operators available, every designed to be sensitive to specific sorts of edges. Edge orientation, noise environment, and edge structure are all factors to consider when choosing an edge detection operator. The operator's geometry defines the typical direction in which it is more sensitive to edges. Operators can indeed be tuned to search for vertical, horizontal, or diagonal edges. Edge identification in noisy pictures is challenging because both the noise as well as the edges include high frequency content (Ahmed S Abdullah, 2019).

**Prewitt Operator**

Edges are computed by subtracting the matching pixel intensities of a picture. All edge detection masks are sometimes referred to as derivative masks. Since that image is likewise a signal, only differentiation can be used to compute changes in a signal. As a result, these operators are sometimes known as derivative masks, derivative operators, or filters.

For edge detection using this method, Prewitt Operator techniques includes two masks, As seen here, one horizontal and one vertical image were convolved with the origin image.

\[
\begin{bmatrix}
1 & 1 & 1 \\
0 & 0 & 0 \\
-1 & -1 & -1 \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
1 & 0 & -1 \\
1 & 0 & -1 \\
1 & 0 & -1 \\
\end{bmatrix}
\]

and using the equation:

\[
\text{Edge magnitude} = \sqrt{IMGX^2 + IMGY^2}
\]

\[
\text{Edge direction} = \tan^{-1} \frac{IMGY}{IMGX}
\]  

(1)  

(2)

**How does it work?**

When researchers apply this mask to the image, it highlights the edges. It simply operates with first-order derivate, calculating the difference in pixel intensities in the edge region. Because the center column, as well as rows of both the vertical and horizontal filters, are equal to zero, it does not contain the actual values of an image, but instead calculates the difference between the right and left pixel values around at that boundary. This increased overall edge intensity and improved it in comparison to the original image.
Colors must be expressed numerically when utilized in digital media such as cameras and computers due to the fact that digital media can only grasp numbers [11]. As a result, color space is a set of principles that permits numbers to be used to describe colors. There are several color representations utilized in digital image processing techniques that are employed in various applications, however the color image within calculator depicts the three fundamental colors of green, red, and blue (Dougherty, 2009).

1. RGB Color Space

The system is amongst the most well-known and well-liked color schemes. It is made up of three main layers: red, green, and blue. Each element in each of these levels necessitates the use of 8 bits (Kaur & Kaur, 2014).

2. YCbCr Color Space

The system is made up of three fundamental layers: the light layer $Y$, the layer $Cb$ (the color layer generated by the difference between both the lighting layer and the blue layer), and the layer $Cr$ (the color layer produced by the difference between the lighting layer and the red layer). This system is represented by the symbol $YCbCr$ in the figure below (s Abdullah, 2020; Thanoon & Alsaiif, 2011).

$Y$ is the color's luma component. The brightness of the color is represented by the luma component. This refers to the color's light intensity (Ahmed Saadi Abdullah, Sleman, & Waleed, 2021).

$Cb$ and $Cr$ are the blue and red components of the chroma component, respectively. That is to say, "$Cb$ is the blue component in relation to the green component." $Cr$ represents the red component in relation to the green component (Waleed, Abdullah, & Ahmed, 2020).

The following equations were used to convert color scheme from RGB into YCbCr:

\[
Y = 0.2126 \times RED + 0.7152 \times GREEN + 0.0722 \times BLUE \\
CB = (BLUE - Y) \\
CR = (RED - Y)
\] (3) (4) (5)
Suggest Algorithm and Result

In this paragraph, we will discuss the basic steps of the proposed system, the essential phases of this system are depicted in Figure 1.

![Figure 1 Flowchart of suggested algorithm](image1)

In First step, the digital color images (RGB) are read, where the reading process is based on an image that available either in the database or captured directly using any imaging device figure 2 show the RGB image.

![Figure 2 Color image](image2)
Then, we resize the image in proportion to the number of processor unit used in the system, and according to the following equation

\[
\text{size}(\text{image}) \mod n = 0 \quad (6)
\]

\( n \) the number of processor unit

For example, if the number of processor unit used in the distributed system is (4), then the divination operation's result of image size by the number of processor unit must be equal to zero, and so on for other options. If the equation condition didn't achieved, the image will be resized again to reach the appropriate size.

In Second step, the color system is changed at this step from the standard (RGB) system to the (ycbcr) system, then disassembled the layers of this system to the basic layers, Layer (y), Layer (cb) and Layer (cr). The (y) layer is used as it represents the light in glayer in this system, as shown in Figure 3.

![Figure 3 Ycber color space](image)

In Third step Digital image processing algorithms are used and applied to the lighting layer (y), where noise removal algorithms are used, and brightness improvement algorithms are used in order to increase the clarity of the image, so the edge detection process is more accurate, as shown in Figure 4.

![Figure 4 Brightness layer](image)
In Fourth step to speed up the processing operation and obtain accurate results with high speed, which are desirable in many applications, the most important such us the medical field as well as the military field. The concept of distributed systems is used in the process of distributing the image to the processors, then each processor unit receives the part assigned to it from the original image, then each processor applies the (prewitt) method to each part to determine the edges of this part of the original image. The simulation was carried out this system based on 4, 8 and 16 stations, as shown in Figure 5.

Fifth step, After the process of edge detections using the (prewitt) method, the parts of the image are collected after being processed by the distributed stations, as shown in Figure 6. The total time is calculated.

From reading the image to the process of assembling the parts of the final image, as Figure 7 shows the details of the time spent in the processing process.
Results After applying the proposed system (DPEDA) on a group of different images (medical, military, natural, personal), the results showed that relying on the concept of distributed systems in this system gave a higher speed and the same accuracy that appears in the absence of the use of the distributed systems concept. Time criterion for measuring system efficiency as shown in Figure 8.
Conclusion

The process of detecting the edges of any entity within the image is one of the basic tasks of cutting that entity from the image and then studying that entity by performing a set of operations on it, whether it is determining the size of that entity, especially in medical topics, a good identification will lead to a good diagnosis and then give The right medicine, also of the important matters is speed in the treatment process, where speed is considered in some cases, such as the medical or military field, one of the most important criteria for the strength of the system, so in this article an idea was given about increasing the efficiency of one of the types of digital image processing operations through the use of the concept of distributed systems. It was studied on more than one type of distributed systems by changing the number of processing units used with the presence of a master node, high-accuracy edge detection was obtained with high speed. Be to a certain extent in order to maintain the accuracy of the edges.
References


