International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.7, No.5 (2014), pp.323-322
http://dx.doi.org/10.14257/ijsip.2014.7.5.28

Detection of Defects on Steel Surface for using Image Segmentation Techniques

S. M. Ramesh, B. Gomathy, T.V.P.Sundararajan

Bannari Amman Institute of Technology, Sathyamangalam, India
rameshsm@bitsathy.ac.in

Abstract

An online surface inspection system m of hot rolled strips is introduced. This system is designed t o detect such main Surface defects on hot rolled strips as scar, scratches, pits, water drops Cracks. Cross hatchings, and so on. Multiple CCD area scan cameras are adopted to capture images of strip surface simultaneously, and all the images are processed by parallel computation system Real-time, which is supported by fast image process techniques and parallel computation techniques, in order to snap main defect regions on the surface of strips. At last, the defects will be classified to several types. The application of the system to practical production line shows that it can detect main defects of hot rolled strips more effectively than traditional method, and it is easily to be maintained.

Keywords: *Digital image processing, Image segmentation techniques, Homogeneity, Region based system.*

1. Introduction

Image segmentation refers to the process of partitioning an image into multiple regions with the goal to simplify and change the representation of the image into something that is more meaningful and easier to analyze. The result of image segmentation is a set of regions that collectively cover the entire image, or a set of contours extracted from the image. Region refers than aggregation of pixels. Properties like gray level, color, texture, shape help to identify regions and similarity of such properties, is used to build groups of regions having a particular meaning. Each of the pixels in a region Are similar with respect to some characteristic or computed property, such as Color, intensity, texture. Adjacent regions are significantly different with respect to the same characteristic(s).

In mathematical sense the segmentation of the image I, Which is a set of pixels, is the partition of Into n disjoint sets R1,R2,...,Uncalled segments or regions such that their union of all regions equals I.

$$I=R1 \cup R2 \cup ... \cup Rn \quad (1)$$

2. Digital Image Processing and Computer Vision

An image may be defined as a two dimensional function (x,y).Where x and y are spatial coordinates and the amplitude of the function f at any pair of coordinates (x,y) is called the intensity of the image at that point. When x, y and the amplitude of values of fare all finite, discreet quantities, the image is called a digital image. The field of Digital Image Processing refers to processing digital images by means of a digital computer.

Vision is the most advanced of our senses and images play a very important role in human perception. However, unlike humans whose vision is limited to the visible band of the electromagnetic spectrum, imaging machines cover almost the entire electromagnetic spectrum. They can operate on images generated by sources that humans are not accustomed to associating with images. These include ultrasonic sound, electron microscopy and computer generated images. This is referred to as computer vision.

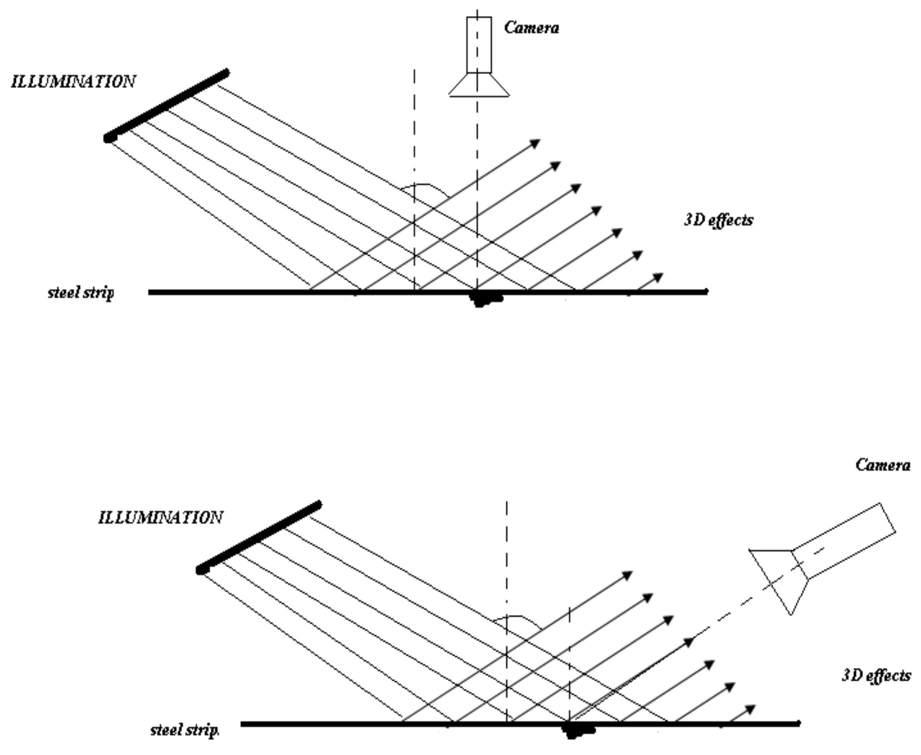


Figure 1. Two different illumination modes

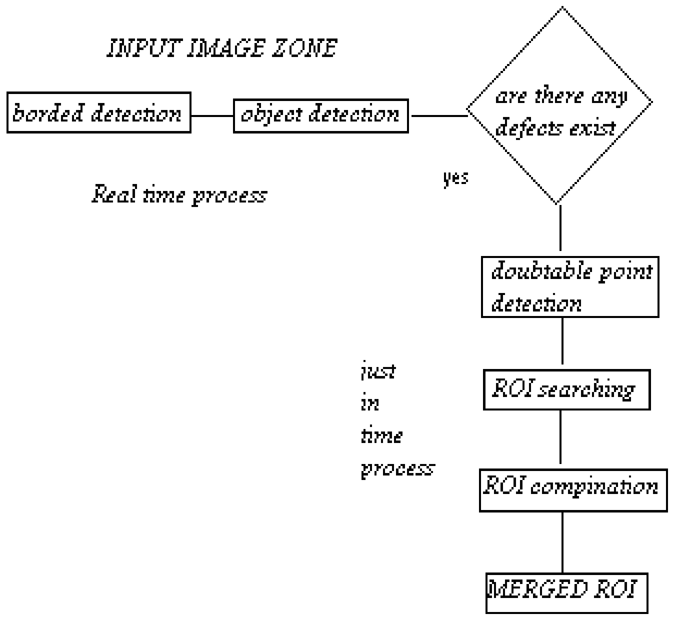
However, it must be noted that there is a very fine boundary line between Digital image processing and Computer vision. A study of image processing and Computer vision is aimed at improvement of pictorial information for human interpretation and processing of image data for storage, transmission and representation for autonomous machine perception.

Image segmentation is a process in which regions or features sharing similar characteristics are identified and grouped together. Image segmentation may use statistical classification, thresholding, edge detection, region detection, or any combination of these techniques. The output of the segmentation step is usually a set of classified elements. If an image has been preprocessed appropriately to remove noise and artifacts, segmentation is often the key step in interpreting the image. Most segmentation techniques are either region-based or edge based.

Region based techniques rely on common patterns in intensity values within a cluster of neighboring pixels. The cluster is referred to as the region, and the goal of the segmentation algorithm is to group regions according to their anatomical or functional roles.

Edge-based techniques rely on discontinuities in image values between distinct regions, and the goal of the segmentation algorithm is to accurately demarcate the boundary separating these regions. Region-based segmentation methods attempt to partition or group regions according to common image properties. These images properties consist of; Intensity values from original images, or computed values based on image

Operator, Textures or patterns that is unique to each type of regions, Spectral profiles that provide multidimensional image data. Elaborate systems may use a combination of these properties to segment images, while simpler system may use a combination of these properties depending of the type of data available.



2.1. Thresholding

Threshold is the simplest way to perform segmentation, and it is used extensively in many image processing applications. Thresholding is based on the notion that regions corresponding to different regions can be classified by using a range function applied to the intensity values of image pixels. The assumption is that different regions in an image will have a distinct frequency distribution and can be discriminated on the basis of the mean and standard deviation of each distribution.

The major drawback to threshold-based approaches is that they often lack the sensitivity and percipiency needed for accurate classification. The problem gets severe in case of multi-modal histograms with no sharp or well-defined boundaries. It is often difficult to define functional and statistical measures only on the basis of gray level value (histogram).

2.2. Region-growing based segmentation

In Region-growing based segmentation, homogeneity of regions is used as the main segmentation criterion. The criteria for homogeneity can be Cray level, color, texture, shape or model. Region rowing approaches is the opposite of the split and merge approach and is a bottom up process. It can often give very good segmentations that correspond well to the observed edges, however starting with a particular seed pixel and letting this region grow completely before trying other seeds biases the segmentation in favor of the regions which are segmented first. This can have several undesirables' effects:

Current region dominates the grow process-ambiguities around edges of adjacent regions may not be resolved correctly. Different choices of seed may give different segmentation results. Problems can occur if the (arbitrarily chosen) seeds point lies on an edge.

2.3. Region splitting

The basic idea of region splitting is to break the image into a set of disjoint regions, which are coherent within themselves. This is a divide and conquers or top down method.

2.4. Merging

If only a splitting schedule is used then the final segmentation would probably contain many neighboring regions that have identical or similar properties. We need to merge these regions. The result of region merging usually depends on the order in which regions are merged.

Table 1. Detection and Recognition Rate of Some Main Defects

Item	Scar	Scratches	Pits	Cracks
Total Defects	965	3436	2898	738
Detection Number	902	3123	2766	678
Detection Rate (%)	93.47	90.89	95.45	91.87
Recognition Number	855	2944	2686	633
Recognition Rate (%)	88.60	85.68	92.68	85.77

2.5. Split-Merge Segmentation

A combination of splitting and merging may result in a method with the advantages of both the approaches. Split-and-merge approaches work using pyramid image representations. Regions are square-shaped and correspond to elements of the appropriate pyramid level. If any region in any pyramid level is not homogeneous (excluding the lowest level), it is split into four sub-regions-these are elements of higher resolution at the level below. If four regions exist at any pyramid level with approximately the same value of homogeneity measure, they are merged into a single region in an upper pyramid level. We can also describe the splitting of the images using a tree structure, called a modified quad tree. Each non-terminal node in tree has at most four descendants, although it may have less due to merging. Quad tree decomposition is an operator that subdivides an image into blocks that contain "similar" pixels. Usually the blocks are squad, although sometimes they may be rectangular. Quad tree decomposition is used in variety of image analysis and compression applicants.

An unpleasant drawback of segmentation squad tree is the square region shape assumption. it is not possible to merge regions which are not part of the same branch of the segmentation tree. Because both split-and-merge processing options are available, the starting segmentation does not have to satisfy any of the homogeneity conditions. The segmentation process can be understood as the construction of a segmentation quad tree where each leaf node represents a homogeneous region, splitting and merging corresponds to removing or building parts of the segmentation quad tree.

2.6. Modern Methods for Image Segmentation

Multi -resolution and multi-channel features
Features fission techniques
Multi-classifier decision combination
HMM, GMM, CRF-and GMRF-based techniques
Artificial Neural Networks-SVM and FFNN
Neuro-fuzzy and soft-computing (SA) techniques
Active contours, watershed transforms
Decision Trees and hierarchical analysis
Probabilistic approaches

3. The Tool

We have used MATLAB as the software tool to accomplish the objective of this project. The algorithm has been developed in MATLAB and later on in visual C++ as well. MATLAB is numeric computation software for engineering and scientific calculations. The name MATLAB stands for MATRIX LABORATORY.MATLAB is primarily a tool for matrix computations. MATLAB was originally written to provide easy access to the matrix computation software packages. MATLAB is a high-level language whose basic data type is a matrix that does not require dimensioning. There is no compilation and linking is done in high-level languages, such as C or FORTRAN. Computer solution in MATLAB seems to be much quicker than those of high-level language such as C or FORTRAN.

All computations are performed in complex valued double precision arithmetic to guarantee high accuracy. Since MATLAB is also a programming environmental user can extend the functional capabilities of MATLAB by writing new modules. MATLAB has a large collection of toolboxes in a variety of domains. Some examples of MATLAB toolboxes are control system, signal processing, neural network, image processing, and system identification. The toolboxes consist of functions that can be used to perform computations in a specific domain. The toolbox which i have used in this project is the image processing toolbox.

4. Downscaling

Downscaling/Downsizing is the scaling down of the image-to-be-segmented to a fraction of its original size. Blocks of n x n pixels are clubbed together to represent one pixel in the scaled down image. Every such pixel is assigned an intensity value equal to the average intensity value of the n x n block it represents.

[n can be any positive integer. I have chosen n to be 4]

This way a N x N image is scaled down to N/n x N/n image an overall decrease in the number of pixels by a factor on n*n.

What downscaling of the image aims at achieving is lesser processing time. As is evident, it does this by decreasing the number of pixels to be processed. But the luxury of faster processing codes at the cost of loss of some amount of accuracy, as far as the final output is concerned. Details within the n x n blocks lose their identity on downscaling. This loss can be minimized by using small value of n. But a very small value of n would fail the purpose of downscaling. Hence I have chosen n to b 4-neither very small nor very large a value. Images with details larger than 4x4 blocks do not run the risk of losing any information. An image that is downscaled before the splitting process has to be scaled back to its original size after the merging process.

5. Spliting

If an image does not meet a certain homogeneity criterion, it is split into four segments of equal size and this process continues recursively. Put in other words, if the criterion for splitting the image is split into four equal segments and then each new segment created is tested for the splitting criterion and the process continues. Theoretically, splitting can continue till segments as small as the size of one pixel is formed... This not only consumes a lot of time but also is undesirable as segments, so small hardly male any sense. Hence, a minimum limit is set beyond which no splitting occurs.

5.1. The splitting criterion

The splitting criterion is used to decide whether and image is image segment should be further segmented or not. The criterion have used is the difference between the maximum and the minimum intensity values for a particular segment, naming is mdev. If this criterion (mdev) is greater than a fixed value then the image segment is split into four equal segments. This fixed value is calculated based on the mean deviation from the mean for the image. This is done by first calculating the mean deviations for the image and then checking which range does this value fall in. accordingly a value is set.

As the program proceeds an array is declared which stores information regarding the various parameters of a segment. Each row in the array, which represents a segment, is checked for splitting criterion. Every time the segment is split, the four new segments created are enter into four new rows of the array and the array keeps growing.

This process continues still a stopping criterion is met. It is a limit put on the size of a segment. As mentioned earlier the stopping criterion prevents segmentation from continuing to the pixel level.

5.2. Detailed description of the code main program

Split-downsize () is declared as a function which when called by another function returns the values of two variables, namely ‘a’ and original. The function execution starts with clearing all the items from the work space, freeing up system memory and closing all windows.

A bitmap (.bmp) image is read into gray1.The rag image is then converted to a grayscale image using the function rgb2gray () and is stored in image. The function mean deviation () (explained later), is called with the array image as its argument. The value it return is stored in mndev. The value in mndev is then checked for the range it falls in and accordingly a value is assigned to split_crit. At this stage, a copy of image is stored in original.

The size of image is determined by using the function size () and the row and column sizes are stored in dx and dy respectively. Gray, an array of zeros, is declared to store the downscaled image. Image is scanned and the averages of 4x4 blocks of it are calculated and stored pixel values of grays and w store the row and column size of the downscaled image .i.e. gray rmin, rmax, cmin, cmax are initialized to the values 1, h, 1, w, respectively. As the names suggest, these variables store the border row and column numbers of an array (an image segment).To start with, the array is gray.

Rmin stores the value of the top most row of the image Rmax stores the value of the bottom most row of the image Cmin stores the value of the left most column of the image.

Cmax stores the value of the right most column of the image. gav is assigned the average intensity vague and gstd is assigned the standard deviation value of gray. The function maxdev( )is called and the values it returns are stored in idev, min and max.

Array a, which stores the parameters of the segments created, is declared. Variables p, q, r are then initialized to 1.

p points to the row of array a that contains the information about the segment that is being processed.

q Points to the row of array that will store the information about the segment that is created next.

The array ‘a’ is updated with the parameters of the first segmented, which is the full image itself. The while loop, which starts after this, is entered only if the segment being processed has a segment height greater than or equal to 1.It is then checked whether mdev is greater than the splitting criterion, split_crit, or not.

If it is found to be greater is incremented by one so as to point new row in array a where the new segment to be created will be stored. The variables rmin, rmax, cmin, cmax are also updated. This is done four times so that four new segments are created. Each of these four times, the variables gavg, mdev, min and max are recalculated based and the array a is updated. After the fourth time this is done, p is incremented by one so as to point to the next segment to be processed.

5.3. Region Continuity

After this splitting and merging process the output that is obtained is a segmented image but the borders of the segments are formed by straight lines and right angles. They do not follow the curves that are present in the input image. The region Continuity process is implemented at this juncture in order to generate a segmented image which reassembles the input image to a greater extent. The region continuity process can actually be considered as a combination of Region Growing and Region Shrinking Process. It is executed in the following manner.

A label matrix is generated from the segmented image after the merging process. The label matrix is scanned for edges by a 2x2 mask. Whenever an edge is detected (it happens only at the segment boundaries), region growing or shrinking is performed at those coordinates at pixel, level, based on the requirement. When a border is detected, pixels on either sides of the border are checked for compatibility with the border pixels, in the original image. This process continues till the number of compatible pixels in the segment on the other side of the border. This mean that the former segment (call it segment A) has grown into the letter segment (call it segment B).At this stage the compatible pixels in segment B are merged to segment A. Thus segment A was grown or put in other words, segment B was shrunk. This process continues till the label matrix is completely scanned and a final output image is generated.

5.4. Merging

Splitting the image gives an output which is n image with a number of segments. Many of these segments may be similar to each other and once need to be merged. This gives rise to the need for the 'Merging'. The merging process is carried out in the following manner.

For a particular segment, all its neighboring segments (i.e. Segments with common boundaries) in question is checked. The Segments are merged if found compatible.

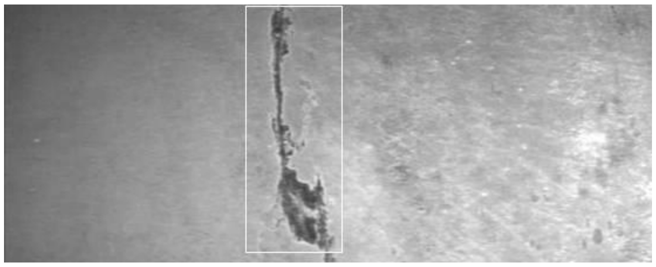


Figure 2. Scar

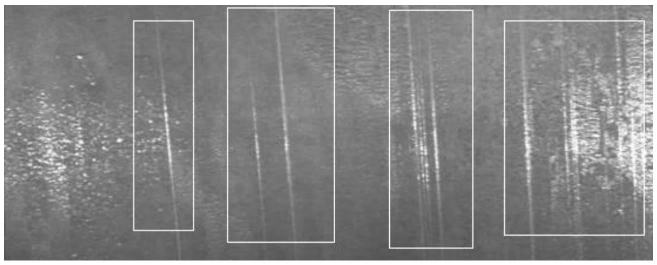


Figure 3. Scratches

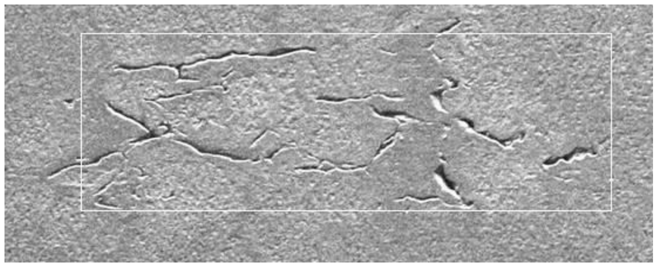


Figure 4. Pits

5.5. The compatibility criterion

The compatibility criterion chosen the 'average intensity value of the segment'. If the difference between the average intensity values of two neighboring segments is less than a prefix tolerance level, the two segments are merged.

5.6. Detailed description of the code

The execution of this program begins with the function call split downsize ().Two values are returned by this function call, namely, the array a which is stored in s and the original image which is stored in original. So, before the merging process begins, we have in readiness, a segmented image and an array containing all the information about the segments.

Constants merge_cri and e are initialized. Their use will be seen later in the program. Copies of the original un-segmented image are made and stored in gray in gray4 and org for possible use later in the program. m and n store the row and column sizes of gray4 and org.

The next step is to generate an array, our of the array s, containing only the final segments and not the segments which were further segmented. Such segments are not difficult to locate as the 8th column of such segments were marked during the splitting process. The task is accomplished by scanning the array s and checking if its 8th column is zero or does it have a certain value. Rows with 8th column zero are ignored and the rest of the rows are copied to a new array. The new rows are copied to a new array. The new array created is named list. A few more columns are also added. One of them stores the area at the segment and the rest are initialized to zero so that they could be used at a large state. The number of rows and hence the number of final segments in list are denoted by q.

At this stage another array 'rec' is created. Every row of this new array would store formation about the merged segments. The first column stores the number of segments that are merged together. this value is updated after every merging operation. the third column is more of a marker. it has a value equal to one of it that particular row is declared redundant. Otherwise, it stores the total size of the merged segment. The fourth column and the columns after that store the information about the segments that have been merged into one. This information which is stored in 'rec' is actually the serial no of the segment in the array list.

To sum up things, before the merging process begins we have two array to work with list is an already filled up arrays whose values will be used during the merging process. rec is an empty array into which values will be written as the merging process proceeds. a third array, 'temp' will come into the picture in a short while. This array would be used to store values, as the name suggests, temporarily.

6. Conclusion

The Split-Merge-Region Continuity method adopted in the development of the algorithm resulted in effective segmentation. The various criteria used at different junctures however hold a lot of importance and control the final output greatly. If there is a scope of improvement in the process, then it is here, in deciding the various criteria controlling the process.

The Splitting process, being the first process being conducted, decides the behavior of the process that follow, to a great extent. While over segmentation at the end of this process may slow down the whole process considerably, under segmentation may cause the final output to drift away from the desired result.

The merging process, which takes the output of the splitting process as its input, forms the longest part of the segmentation process. Both these things point to one thing-choosing the right criteria for splitting.

The region continuity algorithm adds value to the output generated after the merging process. The output is more pleasing to the eyes and lot more intuitive the previous output.

After segmentation has been achieved, the algorithm enters the next important stage which deals with bringing out the defects from the rest of the image. Again, the criteria used for this purpose holds the key. The criteria used by me were based segments having intensity values in the extremes. Area was also taken as a criterion for certain images.

The result was impressive indeed. But the point to be made at this juncture is that the same criteria may not work for all sorts of images. Further research needs to be done in this field to develop a generalized criterion which effectively brings out the defects from the main image and holds true for a variety of images.

The pointer p, which points to the primary segment being processed, is used to scan through the rows of the array list. The pointer j, which points to the secondary segment being processed, also scan through all the rows of the list. The for loop controlling p. hence, for every primary segment being processed. the tenth of column list is used to denote whether that segment has already been entered to ‘rec’ or not-If it has been, then the tenth column contains a value which is the serial no of the row in which the segment is entered in ‘rec’, if not, the tenth contains zero.

References

[1] A generalized suffix tree and its (un)expected asymptotic behaviors, SIAM J. Computing, 22, pp. 1176-1198, 1993.
[2] Asymptotic properties of data compression and suffix trees, IEEE Information Theory, 39, pp. 1647-1659, 1993.
[3] Autocorrelation on words and its applications. Analysis of suffix trees by string-ruler approach (with P. Jacquet), J. Combinatorial Theory. Ser. A, 66, pp. 237-269, 1994.
[4] Average profile and Limiting distribution for a phrase size in the Lempel-Ziv parsing algorithm (with G. Louchard), IEEE Information Theory, 41, 478-488, 1995.
[5] Asymptotic behavior of the Lempel-Ziv parsing scheme and digital search trees (with P. Jacquet), Theoretical Computer Science, 144, 161-197, 1995.
[6] On the average redundancy rate of the Lempel-Ziv code (with G. Louchard), IEEE Information Theory, 2-8, 43, 1997.
[7] A Suboptimal Lossy Data Compression Based on Approximate Pattern Matching (with T. Luczak), IEEE Trans. Information Theory, 43, 1439-451, 1997.
[8] Pattern matching image compression: Algorithmic and empirical results (with M. Atallah and Y. Genin), Proc. International Conference on Image Processing, vol. II. 349-352, Lausanne, 1996; full journal version can be found at Purdue CSD-TR-95-083, 1995 (compressed by gzip).
[9] Average profile for the generalized digital search tree and the generalized Lempel-Ziv algorithm, (with G. Louchard and J. Tang), SIAM. J. Computing, to appear.
[10] Pattern Matching Image Compression with Prediction Loop: Preliminary Experimental Results (with D. Arnaud) Purdue University, CSD-TR-96-069, 1996.