

Using Retinex and SVD Algorithms for Detection of Frayed Edge in Steel Plate

Michal Holíš¹, Martin Plaček¹,
Jiří Dvorský², Jan Martinovič², and Pavel Moravec²

¹ Department of computer science, VŠB – Technical university of Ostrava,
17.listopadu 15, 708 33 Ostrava – Poruba, Czech republic,

{michal.holis, martin.placek}@vsb.cz

² IT4Innovations, VŠB - Technical University of Ostrava,
17. listopadu 15/2172, 708 33 Ostrava, Czech Republic

{jiri.dvorsky, jan.martinovic, pavel.moravec}@vsb.cz

Abstract. This paper describes a method that tries to improve the accuracy of a machine vision algorithm for frayed edge detection in cold-rolled electrical grain oriented steel plate with usage of the Singular Value Decomposition. The algorithm being improved is based on preprocessing the image with the Multi Scale Retinex algorithm, application of the Sobel filter and additional evaluation logic.

1 Introduction

In our previous paper [6] we have presented an image analysis method for detection of frayed edge in cold-rolled electrical grain oriented steel plate. In this paper we enhance this method with usage of the Singular Value Decomposition (SVD) and present results that were obtained with it. The main idea is to use the SVD to preprocess an image that is being analyzed and try to use it to enhance the image so that it improves detection accuracy.

This paper is divided into three main sections. In Sect. 3 short description of the Retinex algorithm and frayed edge detection is described. Section 4 describes the Singular Value Decomposition. In Sect. 5 frayed edge and detection algorithm are described. Section 6 contains information on experiment design and Sect. 7 contains measured results of the experiment. Last Sect. 8 concludes the paper and summarizes achieved results.

2 State of the Art

Detection of the frayed edge itself is subject that has so far not been studied thoroughly. Articles on the origin of this defect and processes involved in creation of it exist but are focused plainly on the metallurgical side of the subject. As State of the Art in this article a selection of image normalization methods is presented.

To enhance the quality of an input image and highlight the frayed edges in the input image many preprocessing methods can be used. In the process of design of our detection algorithm we have tried number of preprocessing method, some of them are described in this section.

2.1 Histogram equalization

The histogram equalization aims to increase global contrast of a processed image to adjust local intensities. This method is useful when an image is composed mainly of close values as it spreads the most frequent values of an image and allows areas with close values to gain much higher contrast. More detailed description of this method can be found in [1].

2.2 Self-Quotient Image

The Self Quotient Image is an extension of The Quotient Image technique first introduced by Riklin-Raviv and Shashua in [11]. This method was first proposed by Wang, Li and Wang in [4].

These methods are both class recognizing methods and they are widely used for an object classifications (for example in face recognition [5]).

Original method (Quotient Image) uses series of bootstrap images to identify ideal illumination free representation of recognized class of objects. Quotient Image of two objects belonging to the same class is then defined as ratio of their albedo functions, thus being illumination free and normalizing lightning conditions and luminance variations.

Self-Quotient Image is extension of previous method that doesn't need training set of images, instead it derives Quotient Image directly from analyzed image. This means, that it can be used purely as image preprocessing method, since no direct knowledge of object's class is required.

2.3 Anisotropic diffusion

Anisotropic diffusion (also referred to as Perona-Malik diffusion) is technique first proposed by P. Perona and J. Malik in [10] that reduces noise and preserves important details of the image that are necessary for correct interpretation of the image. It is based on generating family of parametrized images, where each of these images is combination of the original image and selected filter.

3 Retinex

3.1 Introduction to Retinex

In real life huge difference in color quality of observed scene and detail of recorded image can often be perceived. The most apparent difference is loss of color accuracy and image detail, especially in darker areas covered by shadows. As a result

recorded images often seem dimmed compared to observed scene. This is caused mainly by inability of camera to distinguish between ambient illumination of the scene and reflectance. Illumination is by its nature independent of the scene, so all the characteristics of observed objects are described only by reflected light component. Recorded image is product of these two components and once it has been evaluated, there is no way we can separate these two values and obtain reflectance, which is critical for correct visual representation of the scene, but human eye still seems to be able to do so.

In 1986, Edwin Land [9] proposed image processing method that tries to simulate behavior of human eye and called it *Retinex*. Retinex is a compound of two words – Retina and Cortex - as retina and primary visual cortex are thought responsible for color constancy of final image. Since then the method has developed and can now be considered family of three main techniques:

- Single Scale Retinex,
- Multi Scale Retinex,
- Multi Scale Retinex with Color Restoration (MSRCR).

3.2 Multi Scale Retinex with Color Restoration

MSRCR can be considered most advanced of these techniques and is thoroughly described in [7]. Simply put MSRCR can be described with equation:

$$R_i(x, y) = \sum_{s=1}^N (w_s \log I_i(x, y) - \log [F(x, y) * I_i(x, y)]) \quad (1)$$

Where i is the index of color band of the image, $R_i(x, y)$ is resulting value of pixel (x, y) of i -th color band, $I_i(x, y)$ is value of i -th color band of original image, $F(x, y)$ denotes Gaussian function and $*$ represents convolution.

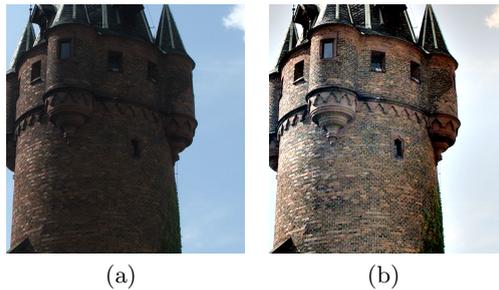


Fig. 1. MSRCR applied to color image. Figure (a): original image, Fig. (b): image processed with MSRCR.

Basically the MSRCR performs set of Gaussian filter operations on input image and computes difference between the filtered and unfiltered image. Each of

is called *singular decomposition* of matrix A and the numbers $\sigma_1, \dots, \sigma_r$ are *singular values* of the matrix A . Columns of U (or V) are called *left* (or *right*) singular vectors of matrix A .

Now we have a decomposition of the original matrix of images A . We get r nonzero singular numbers, where r is the rank of the original matrix A . Because the singular values usually fall quickly, we can take only k greatest singular values with the corresponding singular vector coordinates and create a *k -reduced singular decomposition* of A .

Let us have k ($0 < k < r$) and singular value decomposition of A

$$A = U \Sigma V^T \approx A_k = (U_k U_0) \begin{pmatrix} \Sigma_k & 0 \\ 0 & \Sigma_0 \end{pmatrix} \begin{pmatrix} V_k^T \\ V_0^T \end{pmatrix}$$

We call $A_k = U_k \Sigma_k V_k^T$ a *k -reduced singular value decomposition* (rank- k SVD) (U_0, Σ_0 , and V_0 represent matrices filled with zeros).

Instead of the A_k matrix, a matrix of image vectors in reduced space $D_k = \Sigma_k V_k^T$ is used in SVD as the representation of image collection. The image vectors (columns in D_k) are now represented as points in k -dimensional space (the *feature-space*). For an illustration of rank- k SVD see Figure 2.

Rank- k SVD is the best rank- k approximation of the original matrix A . This means that any other decomposition will increase the approximation error, calculated as a sum of squares (*Frobenius norm*) of error matrix $B = A - A_k$. However, it does not implicate that we could not obtain better precision and recall values with a different approximation.

To execute a query Q in the reduced space, we create a reduced query vector $q_k = U_k^T q$ (another approach is to use a matrix $D'_k = V_k^T$ instead of D_k , and $q'_k = \Sigma_k^{-1} U_k^T q$). Instead of A against q , the matrix D_k against q_k (or q'_k) is evaluated.

Once computed, SVD reflects only the decomposition of original matrix of images. If several hundreds of images have to be added to existing decomposition (*folding-in*), the decomposition may become inaccurate. Because the recalculation of SVD is expensive, so it is impossible to recalculate SVD every time images are inserted. The *SVD-Updating* [3] is a partial solution, but since the error slightly increases with inserted images. If the updates happen frequently, the recalculation of SVD may be needed soon or later.

5 Detecting frayed edges on grain oriented electrical steel

The Retinex as the most appropriate image normalization technique was chosen for preprocessing of images in inspection of quality in grain oriented electrical steel making process.

Frayed edges detection is part of surface quality monitoring system. Goal of the system is to monitor grain oriented electrical steel plate's surface during manufacturing process and to detect set of defects degrading quality of final product.

Steel plate is coiled up into the coils. Approximate length of one coil is 4000 meters.

Steel plate continuously runs through the de-carbonization line and it's surface is monitored by set of cameras from both sides. System then analyses input images for defects in real-time.

One of the most problematic defects to detect is frayed edge. In the input image it appears only as a small deviation in brightness in horizontal direction (see Fig. 3). This type of defect is captured from one side of the plate only (as it is visible from both sides) using monochrome digital camera. Resolution of one image is 2400×600 pixels. Width of area captured by one camera is approximately 0.5 m, which means that each millimeter of captured area is represented almost by 5 pixels in final image. Images are automatically archived so they can be worked with to improve quality of defect detecting algorithms and now we currently have base of more then two million test images.

Frayed edges arise on a plate because of insufficient MgO powder coverage of the edges. In the annealing process the uncovered edges are stuck together, because of high annealing temperature, and the defect is formed on a plate when it is unwinded on the next processing line and stuck edges are torn off.

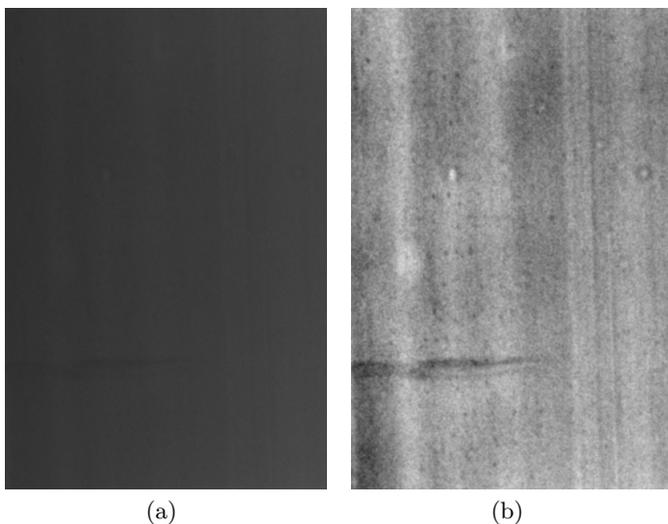


Fig. 3. Example of frayed edge. On Fig. (a) we can see original image, on Fig. (b) is the same image processed with Retinex.

Common edge detecting algorithms used on non-preprocessed images do not provide any meaningful results because frayed edge appears only as very small deviation in input image and is suppressed by noise that is introduced into the image due to low exposition time requirements and environmental conditions that do not allow for better lighting of the scene.

To highlight our area of interest – the frayed edge – and to suppress the light non-constancy is the core of the problem.

Many preprocessing algorithms were tried before the Retinex was chosen for this problem. Among others these light normalization algorithms were tried out: Histogram Normalization [1], Self Quotient Image [4], Anisotropic Diffusion [10] and many more.

6 Experiment Design

In our experiments we will try to detect frayed edges on plate using simple Sobel filter [8] that will be applied after all preprocessing algorithms. Results of this Sobel filter and some additional processing (filtering edges caused by noise ...) will allow us to judge quality of used preprocessing algorithms.

SVD's result will be used as a mask on preprocessed image. This will allow us to perform detection only on areas highlighted by the SVD. First we will try to apply this mask to original input image, so we can see if the SVD itself is able to replace the Retinex (if the SVD's mask is accurate enough then the Sobel filter might yield correct results). Additionally we will try to apply mask to image already preprocessed with the Retinex to see whether it can improve accuracy of the Sobel filter. Finally we will run the Sobel filter on image processed purely the Retinex for reference.

SVD image can be computed either from input image or from image already processed by the Retinex algorithm. Both variants will be tried in our experiments.

As test database life data captured with system described in Sect. 5 will be used. Database consists of 100 images containing Frayed Edge on left edge of the steel plate.

All relevant parameters of preprocessing algorithms used for experiments are summarized in Table 1. First column *Name* of the table contains identification name of the run. Second column *SVD source* specifies whether the SVD was computed from original image or from image with the Retinex applied. Third column *Source image* specifies whether the SVD's mask will be used on original image or on image with the Retinex. Column *Multiplier* contains value that will be used to multiply all values in resulting image to enhance the brightness of the image. Column *Lower bound* contains lower bound constant, all values lower than this bound will be clipped to 0. Last column *Upper bound* contains upper bound constant, all values higher then this constant will be automatically adjusted to 255. First row of the table represents reference algorithm run.

First we will run our reference algorithm and store locations and depths of frayed edges present on the image. Then all the other settings will be run and their results will be compared to those of the reference run.

To conclude contributions of SVD to the detection the following metric will be used:

- if the Sobel filter is able to detect at least 80 % of frayed edge's length, then the detection is considered as successful,

Table 1. Parameters of experiments.

Name	SVD source	Source image	Multiplier	Lower bound	Upper bound
Retinex	-	-	-	-	-
SVD 1	Original image	Original image	100	200	200
SVD 2	Original image	Retinex image	100	200	200
SVD 3	Retinex image	Original image	100	200	200
SVD 4	Retinex image	Retinex image	100	200	200
SVD 5	Original image	Original image	100	150	225
SVD 6	Original image	Retinex image	100	150	225
SVD 7	Retinex image	Original image	100	150	225
SVD 8	Retinex image	Retinex image	100	150	225

- if the algorithm detects false frayed edge of length at least 10 % of the image’s size, then it is considered as false positive,
- otherwise detection is considered unsuccessful.

Ratio of successful detections and sum of false positives and unsuccessful detections will then be our metric that will allow us to compare individual settings. This metric is described in Eq. (2).

$$m = \frac{s}{s + f + u} \cdot 100 \% \quad (2)$$

Where:

- m - final number specifying accuracy in percents of given run, the higher the number the better,
- s - number of successfully detected images,
- f - number of false positive detections,
- u - number of unsuccessful detections.

7 Experiment Results

Table 2 summarizes results we have obtained. Structure of the table is similar to Table 1 only column *Success rate* is added. This column contains result of the experiment, how this number was obtained is described in Sect. 6 and in Eq. (2).

We can see that the SVD by itself was not able to highlight defected areas sufficiently. All runs that were detecting the defect from original image failed to detect single defected image from the testing set.

Runs that used the Retinex image as source image for detection were able to detect defects with some success. Those that used the Retinex as source both for the SVD and for detection performed much better. Those that used the Retinex only as source image and SVD mask was computed from original image performed unconvincingly. This is caused by fact that SVD highlighted only

Table 2. Experiment results.

Name	SVD source	Source image	Multiplier	Lower bound	Upper bound	Success rate
Retinex	-	-	-	-	-	100 %
SVD 1	Original image	Original image	100	200	200	0 %
SVD 2	Retinex image	Original image	100	200	200	0 %
SVD 3	Original image	Retinex image	100	200	200	16 %
SVD 4	Retinex image	Retinex image	100	200	200	91 %
SVD 5	Original image	Original image	100	150	225	0 %
SVD 6	Retinex image	Original image	100	150	225	0 %
SVD 7	Original image	Retinex image	100	150	225	42 %
SVD 8	Retinex image	Retinex image	100	150	225	91 %

significantly different areas in the image and omitted the less distinctive ones thus shortening the length of detected defect.

Sample images of all settings can be found in Fig. 4. We can see original image 4(a), Retinex image 4(b), SVD (upper bound: 200, lower bound: 200) computed from original image 4(c), SVD(upper bound: 200, lower bound: 200) computed from the Retinex image 4(d), SVD (upper bound: 225, lower bound: 125) computed from original image 4(e) and SVD(upper bound: 225, lower bound: 125) computed from Retinex image 4(f).

From the results obtained we can see that the SVD itself led to no improvements in detection. As an algorithm to highlight defected areas the SVD itself failed and even with help of the Retinex algorithm it was not able to achieve 100 % success rate, so the addition of the SVD will not be an improvement over the current the Retinex solution and will not help to detect defects that the Retinex previously was not able to.

8 Conclusion

In this paper we have tried to use Singular Value Decomposition to improve accuracy of frayed edge detection. Proposition that the SVD itself might be able to successfully highlight defected areas has proven to be wrong as it was not able to correctly detect single frayed edge in test images.

When used in combination with the Retinex the detection algorithm was able to achieve 91 % success rate of reference the Retinex algorithm. This means, that with addition of this mask the algorithm performed worse then without it. Our hopes were that with usage of the SVD mask the detection algorithm will detect defects in all test images and we might be able to lower the threshold on edge detection algorithm thus finding more subtle frayed edges and improving the accuracy of the algorithm. With success rate of 91 % this idea is proven to be wrong.

To summarize results presented in this paper – method introduced in our previous paper [6] still achieves best results we were able to obtain so far. Usage

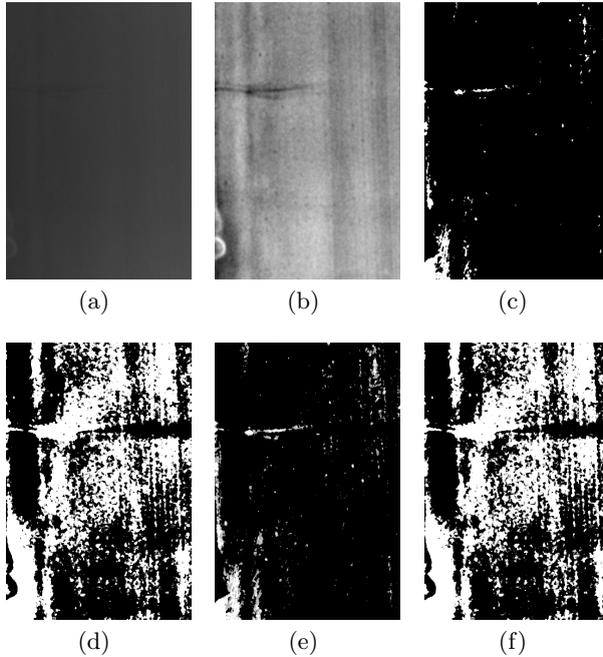


Fig. 4. (a) - Original, (b) - Retinex, (c) - SVD1, (d) - SVD3, (e) - SVD5, (f) - SVD7

of the SVD both as a pure defect detection algorithm or as a mask only lowers success rate of the detection and makes usage of the SVD in our algorithm pointless.

In our future work we would like to test more lighting normalization methods and try to improve detection accuracy so that the algorithm would be able to reliably detect even more subtle frayed edges.

Acknowledgment

This work was supported by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070) and by the Development of human resources in research and development of latest soft computing methods and their application in practice project, reg. no. CZ.1.07/2.3.00/20.0072 funded by Operational Programme Education for Competitiveness, co-financed by ESF and state budget of the Czech Republic.

References

1. Acharya and Ray. Image processing: Principles and applications. *Wiley-Interscience*, 2005.

2. M.W. Berry and M. Browne. *Understanding Search Engines, Mathematical Modeling and Text Retrieval*. Siam, 1999.
3. M.W. Berry, S.T. Dumais, and T.A. Letsche. Computation Methods for Intelligent Information Access. In *Proceedings of the 1995 ACM/IEEE Supercomputing Conference*, 1995.
4. S. Li H. Wang and Y. Wang. Face recognition under varying lighting conditions using self quotient image. *IEEE International Conference on Automatic Face and Gesture Recognition*, 2004.
5. Guillaume Heusch, Fabien Cardinaux, and Sebastien Marcel. Lighting normalization algorithms for face verification. 2005.
6. Michal Holis and Martin Placek. Detecting frayed edge in steel plate using multi scale retinex algorithm. In *Wofex*, pages 440–446. VŠB TUO, 2012.
7. D.J. Jobson, Z. Rahman, and G.A. Woodell. A multiscale retinex for bridging the gap between color images and the human observation of scenes. *IEEE Transactions on Image Processing*, 6:965 – 976, 1997.
8. N. Kanopoulos. Design of an image edge detection filter using the sobel operator. *Solid-State Circuits, IEEE Journal of*, 23:358 – 367, 1988.
9. E. Land. An alternative technique for the computation of the designator in the retinex theory of color vision. *Proc. Nat. Acad.*, 83:3078–3080, 1986.
10. P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7), 1990.
11. A. Shashua and T. Riklin-Raviv. The quotient image : Class based re-rendering and recognition with varying illuminations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2), 2001.