7th International DAAAM Baltic Conference
"INDUSTRIAL ENGINEERING "
22-24 April 2010, Tallinn, Estonia

IMPROVEMENT OF TIMBER CODE READING METHODS

Juurma, M.; Tamre, M.

Abstract: *In this paper the code marking and reading system is discussed that was created for tracing logs as a part of the EU 6th Framework project Indisputable Key. A set of images of codes on logs was acquired in SK Rolpin (SKRP) sawmill in France and an automatic code reading software was developed and initial readability analysis was performed. Machine vision and decoding algorithms were developed and implemented. Code reading algorithm consists mainly of four distinct steps: image preprocessing, image orientation and compression, particle analysis and decoding of the code. This marking and decoding system was created to prove of values that can be gained thorough marking individual logs in sawmill. In parallel two different log code marking schemes were used and compared. Efficiency analysis of the image processing algorithm reveals clear advantages of distinct and application specific algorithms which are analyzed and demonstrated in the paper. In spite of efficiency of the algorithm there exist several methods to increase artificially code contrast and visibility in different conditions. One of code visibility enhancing methods is using externally activated paints for code marks, especially infrared or ultraviolet light activated inks. Performance of this kind of markings is analyzed. Respective code marks are applied to logs and readability is tested. The test results are discussed in this paper and future directions for high effective code marking are given.*

Key words: timber, coding, computer vision, fluorescent, traceability, wood industry.

1. INTRODUCTION

The Indisputable Key project's main objective is to develop tools and knowledge to enable a significant increase in raw material yield and in utilization of production resources in wood industry conversion chain, thus decreasing the environmental impact [1].

The code used in SKRP is a binary line type code. The code consists of orientation marks and code elements. Code elements are arranged to eight by three grid. Every row starts with an orientation mark followed by three code elements. Every code element corresponds to a numeric value.

The codes were printed onto the logs using an ink-jet printer with blue ink and with ink consisting of Luminescent Nano-pigment (LNP) material. LNP are inorganic ceramic crystals with luminescent properties [2].

For image analysis and algorithm developing we used National Instruments Labview and its Vision Development module.

2. IMAGE ANALYSIS

2.1. Analysis of image color planes

Analysis of image planes is done to find out if all of the color planes of SKRP images have equally necessary amount of information. The SKRP code has a dominant color; in the case of coding with normal color the dominant is blue, in case of fluorescent coding the dominant color is green. Nevertheless it is known that in fluorescent marking LNP emits also in blue part of the light spectra.

In color marking case the dominant in the code is as mentioned, blue. The logs were lighted with an unknown halogen light. A common halogen light has spectra peaks (Fig. 1.) in red (650 nm) and near-infrared area (950 nm) of the spectra.

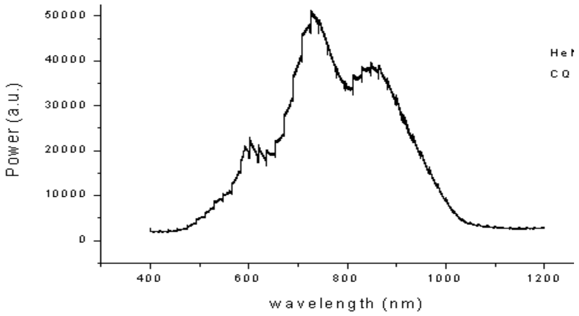

Fig. 1. Spectra of a common halogen lamp [3]

This situation results in bright field effect, where in the image red plane the background is bright and the code is dark. The blue plane should have inverse information of red plane. Green plane consists mainly of miscellaneous information.

In fluorescent marking the dominant color is green, because upon excitation LNP emits green light. The code was illuminated with infrared (IR) light, which was blocked. The images were captured in a dark environment (dark background) which resulted in images that have the most of information (bright code) in green plane. In addition, there is some information of the code in blue plane, because LNP has a weak blue emission.

First, we try to quantify the information in different image planes. In order to compare image planes we use logical difference to measure the information differences in image planes.

$$Dst(x,y) = SrcA(x,y) \wedge \neg\big(SrcB(x,y)\big) \quad (1)$$

Where Dst is the destination image, SrcA and SrcB source images that will be compared. This logic operator keeps bits found in image SrcA that are absent from image SrcB [4].

On fluorescent code images it is evident that the most of information is on the green plane of the image. We choose the green plane as the reference plane and use it to compare with others to see it and decide which information can be transferred to destination image.

The test was done with ten randomly picked images from the real image database. All the images were rotated to horizontal and the code area was masked to remove background objects. Finally, the logical difference operator was applied to keep the bits in the plane under inspection that were missing in green plane.

The histogram of the green plane of each image looks similar every time. There are two distinct peaks; one peak in the dark pixel value (background) and one in bright pixel area (code) whereas pixel values in between are rather equally distributed.

The result images contain very little of visible information, which is the reason we compare the result image histogram with the reference image histogram. It appears that red and blue image planes do not contain bright intensities that are not already present on green plane. As the code is bright on a dark background we can conclude that red and green image planes are not necessary for further image processing in case of fluorescent code images.

We assumed in the beginning of the chapter that the blue plane of color code images have the most of the information. On closer inspection it appears that red plane has the best difference between the code and its background. The halogen lamp has overexposed the background of the code which compared to blue plane is more uniform and creates more contrast with the code.

For image testing with logical difference the images were prepared similarly as in fluorescent marking case. Only difference was that the red plane of the image had to be inversed so that the background would be dark and the code bright.

The logical difference image of blue and red planes has a histogram that largely is similar to the blue image plane histogram. This is because of the background in the blue image plane is dominant in the result image and the bits that are missing in the red image plane are mostly bits that represent the noisy background of the blue image plane. Visual inspections of the result images confirm the latter. It can be concluded that the blue image plane does not contain a significant amount of information that would complement the code on the red plane of images.

2.2. Analysis of image elements

For code element analysis we first look at the profiles of the code elements in the image and also review the histograms of the images.

Profiles of fluorescent mark elements (Fig. 2.) show that normally the elements have maximal intensity values with some discontinuities.

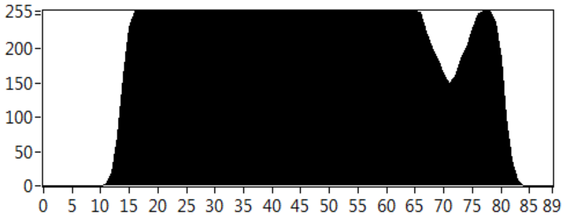

Fig. 2. Example of a fluorescent code element profile

The histograms of the images display one high peak in dark pixel values and another peak in light pixel values (Fig. 3.). The pixel values in between do not have much weight compared to dark and light pixels.

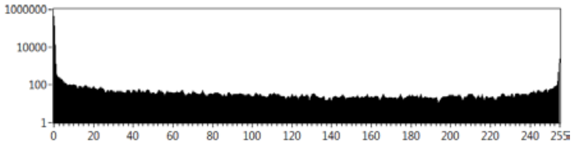

Fig. 3. Example of a fluorescent code histogram

Considering the observations we can conclude that fluorescent code images need image smoothing, preferably in the horizontal direction of the code.

Examination of the color code elements reveal that the profiles are significantly different compared to fluorescent code elements. It appears that element profiles do not have maximal intensity values (Fig. 4.). Instead the bright element values tend to be in the middle values of pixel intensities.

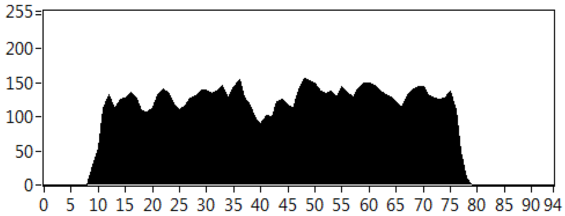

Fig. 4. Example of a color code element profile

The previous statement is confirmed when looking at the image histograms (Fig. 5.). Pixel intensity values end at the middle of the histogram.

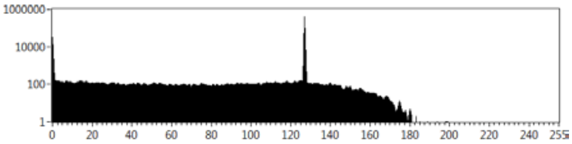

Fig. 5. Example histogram of a color code image

In conclusion, the color code images need histogram modifications in preprocessing to stretch the values to maximum and also image smoothing to reduce discontinuities in the code elements.

3. VISION ALGORITHM FOR CODE IDENTIFICATION

3.1. Introduction

The chapter introduces the reader software developed using National Instruments (NI) LabView Vision package for the RolPin advanced code described in chapter one. The software has been built for online as well as for offline code reading from images taken by the camera. Additional options are introduced allowing testing different image processing options and filters which might be needed for on-site reader implementation.

The description of the algorithm has been divided into five stages:

1) preprocessing and image rotation
2) code edge and direction
3) Code extraction from image
4) Code element extraction (binarization)
5) Decoding

3.2. Image preprocessing and rotation

Due to some distinct properties of NI LabView functions and also to simplify the decoding process, the input image is rotated so that the code is orientated horizontally.

The input image is a 32-bit RGB image. Most of the code information is located in one of the image planes. In case of the LNP in the code radiates green light and in the case of normal code the image plane that has the most of information is red plane. First step is to isolate the image plane. For smoothing the image we use IMAQ Correlate function [4]. This function finds the cross correlation of input image and a template which consists of NxN pixel white square surrounded by black border. A template of 5x5 white pixels and 1 pixel black frame seems to have the best effect of smoothing.

Next, the image is binarized. Code elements tend to be blurred on most of the images. To separate the elements we create two copies of the image. One image is processed with a FFT masking routine to separate the code elements. Finally the image is threshold. The second image is threshold with image background correction function and then dilated. Next, the images are subtracted to remove the code from the image. Resulting image from latter is finally subtracted from the FFT filtered image, removing most the particles on the background of the image.

To find the orientation of the code a rough particle filter is applied which removes particles that do not have the pixel area that is distinctive for the code elements. The final step of the stage is rotating the image clockwise to horizontal position using mean angle value of particles on the image.

3.3. Code orientation edge and direction finding

Code orientation edge is a mark that indicates the direction that the code should be decoded. To find the edge the mean of particle coordinates we used to rotate the image in stage one is calculated. This point is used to create a region of interest (ROI) over the code area. To locate the edge a smaller ROI with the height of search area ROI is moved across the search area with a step of 10 pixels. IMAQ Light Meter function is used to measure pixel intensities in the sliding ROI. Code edge location is the coordinate that has the highest intensity value. In case there is no edge in the search area, “error 1” message is returned.

Code decoding direction is determined by comparing the Y-coordinate of located code edge and coordinate of mean code element particles Y-coordinate.

3.4. Code image extraction

In current stage the code is extracted from the original image. The ROI of extraction is constructed using information gathered from the previous stage:

- Extraction Y-coordinates are the Y-coordinates of code edge search area
- X-coordinates are calculated by using the edge location and constant that represents the width of the code in pixels.

3.5. Code particle extraction

To extract code particles the image has to be binarized. For binarization the extracted image goes through the same procedure as the original image in stage one.

After pre-processing particles are filtered by area. Then the particle center of mass coordinates is found. The coordinates are used to create two histograms; one histogram classifies y-coordinates to eight bins and the other classifies the x-coordinates to three bins.

The x-coordinate histogram is modified so that it consists of roughly histogram bin transition values. Those values will be used in the decoder to discern the code rows.

3.6. Decoding

The decoding routine uses particle coordinates to determine the bits in the code. The first step is to convert the spatial coordinates to binary values. For that, the array of coordinates is sorted ascending by y-coordinate values. Next, every x-coordinate is compared to the threshold array value and also to corresponding histogram value. Those comparisons determine the location of the code element in resulting 2D binary value array that represents the code on image. The next step is to convert the two-dimensional binary array to numerical code. For this the binary array is multiplied with a numerical value matrix array that incorporates the numerical value of every bit on the code.

The array is multiplied 1) with straight code value array; 2) with rotated code value array. Finally, the array members are summed to find the code numerical value. The choice of which value to use is determined by the code direction value that was identified in stage two.

4. EFFICIENCY OF THE ALGORITHM

In current chapter we look at the performance of the algorithm using available image database that was acquired on the SK RolPin production line. The algorithm was evaluated stage-by-stage manually. In each stage failed images were recorded and removed from processing before evaluating the next stage. First we look at the fluorescent code reading and then also color code reading. Images that did not pass the stage at hand were considered as erroneous and removed from test samples.

In the first stage of the algorithm two images of 107 failed to be rotated to horizontal. The reason in one case was absence of code on the image and the other error was unknown anomaly.

In the second stage there were two types of errors. One error was that there was not enough information of orientation edge on the image (13 errors). The other source of errors was caused by recognizing a wrong set of particles as the code edge element (4 errors).

In the third stage in one case the algorithm failed to extract the code area. The reason was “Scale error” that states the code was considerably “bigger” than the other codes. Because of that part of the code was lost during extraction.

In stage four the amount of seven images had errors when the image was binarized and code elements extracted. In each of the seven images one code element was missing from the final code form.

For the final stage of decoding 82 images of 107 were considered correct. All of the codes on the images were correctly decoded.

The algorithm was also similarly tested with color code images. Although the code printing technology and camera setup was similar, the results of that were rather poor; reading success was lower than 20%. The reasons for the poor success are unknown at the moment and need further investigation.

In conclusion, the algorithm provides readability 75% for the first test sets of the fluorescent code images taken in RolPin production line. However, the readability was achieved without any on-site adjustments and in conditions where the marker had problems with some markings and therefore the result is considered promising to achieve close to 100% readability if the on-site testing and adjustments could be possible. Unfortunately SK RolPin production was sharply reduced and any testing at the moment is not possible due to company’s economic situation.

5. OPTIONS TO INCREASE CODE CONTRAST

The more traditional way to enhance colors of objects on images it to use Maxwell’s triangle method for selecting color filters. To enhance the color needed two of its adjacent color filters can be used [5].

Another way to enhance the objects is using luminescent marking which is rather common in microscopy. Luminescent marking enables us to significantly reduce the influence of code background on the image.

Wood marking with phosphorescent dyes was investigated in EU project called Lineset. The conclusions of their tests stated that using phosphorescent dyes is hindered by the dye's long excitation time [6]. Nowadays nevertheless, there are new dye materials that have shorter excitation times.

Fluorescent dyes have short excitation span and thus are more suitable for marking and reading applications. One option of fluorescent material, the LNP, was described earlier.

6. CONCLUSION AND FUTURE WORK

In this paper we have presented an image analysis of log code images tested in SK Rolpin, France. The analysis revealed that the use of fluorescent code marks is a perspective approach in marking products or materials for attaining traceability. The code images for log marking have the most of information on green image plane. In addition we noted that the code elements tend to have good contrast but need horizontal smoothing.

On the other hand, color code images appeared to have the most of information on red plane. In addition, it appeared that the images need histogram modifications and also horizontal smoothing to remove discontinuities.

Efficiency analysis concluded that the vision algorithm for automatic code reading had 75% success rate with fluorescent images and very poor success with color code images.

We can conclude that there is evidence that using luminescent materials to increase code readability with visual marking is feasible. Nevertheless, how and what materials should be used is rather vague knowledge. We propose for the next steps of the work to identify fluorescent materials that can be used for code marking and investigate their usability for visual marking.

7. ACKNOWLEDGMENTS

This work was supported by 6th Framework IST Project No 34732 „Indisputable Key“, Doctoral School of Energy and Geotechnology II, and Estonian Ministry of Education and Research Project SF0140113Bs08.

8. REFERENCES

1. Uusijärvi, R., Annex I - Description of Work 2008-10-21: Detailed implementation plan for next 18 months for month 19-36. 2008.

2. Juurma, M, Tamre, M. An image processing algorithm for decoding line type code. In *8th International Symp. Topical Problems in the Field of Electrical and Power Engineering (Lahtmets, R., ed.).* Tallinn University of Technology, Tallinn, 2010.

3. Peterson, M.A., Halogen light Spectrum, the Fluorescent Spectrum, Web Page, http://tinyurl.com/4bu82 , 4th of Mar, 2009

4. National Instruments Vision for LabVIEW Help. National Instruments. Austin, Texas, USA, 2009.

5. Bolhouse, V. (2009). Fundamentals of machine vision: understanding optics and light. [Online] Vision systems Design webcast, 10th of Feb, 2009

6. Seidla, A. Automated marking of logs: Licentiate thesis. KTH Stokholm, Rootsi, 2004.

9. ADDITIONAL DATA ABOUT AUTHORS

MSc. Märt Juurma
Doctoral student
Tallinn University of Technology
Department of Mechatronics
Ehitajate tee 5, 19086 Tallinn, Estonia
mart.juurma@ttu.ee
+372 620 3207

PhD. Mart Tamre
Tallinn University of Technology
Department of Mechatronics
Ehitajate tee 5, 19086 Tallinn, Estonia
mart@staff.ttu.ee
+372 620 3202