



MASTER THESIS

DESIGN OF A MACHINE VISION BASED FAULTY BEAD WIRE DETECTION SYSTEM

Yvo Delaere
y.delaere@student.utwente.nl

University of Twente
Faculty of Electrical Engineering, Mathematics and Computer Science
Data Management & Biometrics
P.O.-Box 217
7500 AE Enschede
The Netherlands

EXAMINATION COMMITTEE
dr. ir. L.J. Spreeuwiers
J. L. Moreira, PhD
ing. G.J. Laanstra

EXTERNAL SUPERVISOR (Apollo Tyres Ltd)
ir. M. Ter Haar

apollo
TYRES

UNIVERSITY OF TWENTE.

Design of a Machine Vision Based Defect Bead Wire Detection System

Yvo Delaere

Department of EEMCS

University of Twente

Twente, The Netherlands

Email: y.delaere@student.utwente.nl

Abstract—Technological innovations are key to be ahead of your competitors in the tire manufacturing industry. This research proposes a prototype tire bead defect system, which can greatly reduce costs and wasted materials. Machine vision technology is utilized and expanded in the design and realization of a prototype of this system. Images of the tire beads are acquired using a Raspberry Pi based 2-camera system without adding an extra step in the production process. These images are rectified and fused into a top view image using a novel single image calibration approach. The width of the bead is measured using the generated top view image, and beads are classified based on this information. Existing technologies and methods are used in a smart way to construct a low budget solution. Initial results show the potential of the system and show areas where the system can be improved.

I. INTRODUCTION

The global tire manufacturing industry produces over 1 billion tires a year, produced in over 450 tire factories across the world. As in every industry, innovation is key to compete against your competitors. A modern radial tire consists out of different parts, see figure 1. One of these components is the bead, which is a single wire steel ring that keeps the tire on rim once the tire is inflated. During the manufacturing process, beads with defects should be detected and consequently removed. Detecting these defects in the beads as early as possible minimizes the amount of wasted material, and therefore costs. In the current situation, the beads are reinforced by an operator, which applies a wrapping material to the bead such that the probability of a defect is minimized. A time-consuming process, which adds an extra step in the manufacturing process of the tire.

The goal of this project is as follows: ***"Construct a prototype system which can automatically detect defects in tire beads with high accuracy, which fits seamlessly within the existing manufacturing process."***

To reach this goal, answers to the following research questions should be found:

- 1) What defects in the bead must be detected and what are their characteristics?
- 2) What is the most suitable moment in the manufacturing process to detect the defects and their characteristics?
- 3) Given what the defects are and where they can be detected, which machine vision techniques can be used to detect them?



Fig. 1. Cross-section of a radial tyre: tread (1), subtread (2), sidewall (3), NOH nylon overhead (4), belt layers 1 and 2 (5), belt rim tape (6), carcass layer 2 (7), carcass layer 1 (8), bead filler strip (9), bead (10), inner liner (11) and rubber rim band (12).

- 4) What specifications should the data satisfy for reliable detection of the defects in the bead?
- 5) Which constraints should be taken into account in the design of an acquisition system that produces this data?
- 6) What is the design of an acquisition system which produces data satisfying the specification for reliable detection of the defects, while taken the constraints into account?
- 7) How to evaluate the performance of the complete defect detection system?
- 8) What is the performance of the prototype defect detection system?

The highlights of this research are the use of open source-based hard- and software to construct a prototype system that shows great potential for further development into an industrial quality defect detection system. Also, a flexible calibration algorithm is presented, which requires only one image of a checkerboard calibration pattern.

This paper is structured as follows. First, related machine vision applications are discussed in section II, which are used to establish the solution direction. Section III provides a case analysis, in which the environment is described, the defects are analysed and specifications are derived which are listed in table I. In section IV, the proposed design is introduced in a general way using figure 5. Sections V, VI and VII describe

the proposed method in detail. Conducted experiments and their results are described in section VIII, which are discussed in section IX. Finally, the paper is concluded in section X.

II. RELATED WORK

In the tire industry, X-ray imaging is often used to inspect the tires once they are fully manufactured. In [1] and [2], the authors use convolutional neural networks to detect bubbles and impurities in X-ray images taken of images after they are fully manufactured. In this research, we aim to detect defects in a single component of the tire, the tire bead. Doing so enables more precise detection of the defect and simplifies the process of finding where the defect occurs in the manufacturing process. Many machine vision applications exist which can give insights and ideas in which a solution for bead defect detection can be found.

Since tire beads are circular, inspiration for detection strategies can be found in other machine vision applications inspecting circular objects. In [3], an acquisition and lighting system is proposed which captures three bearings in one image. Bearings are segmented and separated from each other using blob analysis. The segmented bearing is then converted to a rectangular image using a polar to cartesian coordinate transformation. Defects such as deformations, rust and scratches are then extracted using pixel intensity characteristics.

In [4], the authors make a clear distinction between dimensional defects and surface defects occurring in polyurethane (PU) sealing rings. They detected dimensional defects, such as burrs and deformations by using a deviation of radius inspection (DOR) method. For every pixel of the cylinder contour, the deviation between the distance to the center point for that pixel, and the mean radius of all contour pixels is calculated. When this deviation exceeded a threshold, the contour pixel was marked as abnormal. Localization of the center point and radius of the cylinder was performed by least-squares fitting of a circle to the contour points. Instead of using least-square fitting of a circle, circular objects can be detected using the circular hough transformation [5]. The advantage of this approach is its robustness for noisy images and occluded circles.

Many variants of hough circle detection exist, but their basic concept is the same. Edges pixels are extracted from the image,

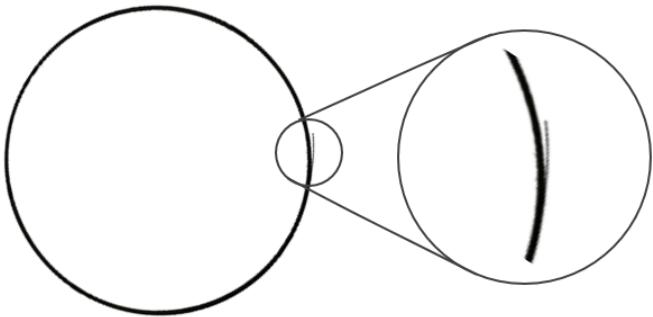


Fig. 2. A typical defect in a tire bead

and a circle of a radius R is plotted around the edge pixel. This is done for every edge pixel in the image. Locations where many plotted circles intersect indicate possible center points (c_x, c_y) of circles present in the input image. The circle is modelled as in equation 1, meaning that the parameters space is 3 dimensional and defined by (c_x, c_y, R) .

$$(x - c_x)^2 + (y - c_y)^2 = R^2 \quad (1)$$

Many improvements on this basic concept have been proposed [6], [7]. These improvements are based on improving computational efficiency and improving robustness against noise.

III. CASE ANALYSIS

This project is performed as an assignment for Apollo Tyres Ltd. The company opened a new production plant in Hungary in April 2017 with state of the art machinery. The company works on technological innovations to improve production efficiency and reducing costs. Given this vision, the company would like to have a low budget solution of around 1000 euros to detect defects in tire beads. In this section, a case analysis is performed, aimed at deriving the information required for constructing a prototype defect detection system. The result of this case study is summarized in table I.

A. Bead analysis

The aim of this step in the design procedure is to determine what should be detected exactly by the defect detection system. The bead is a metal ring, surrounded by a rubber coating. Beads are made from a long, single wire, in which the wire is wound onto a drum by a winding machine. The windings are created in multiple layers, moving from left to right and from right to left. The windings are held together by the cohesive force of the rubberized windings pushed together. Incorrect configuration of the winding machine and human interaction of the beads can cause part of a winding to get separated from the bead itself, causing a 'wild wire' defect. Figure 2 shows an example of this.

The dimensions of the bead are determined by their inner diagonal and the number of windings in the horizontal- and vertical direction. The minimal outer diameter of the bead considered in this project is 340mm, the maximal outer diameter 500mm. The diagonal of a single rubberized wire is 1.1mm. The bead is ring-shaped, meaning that the shape of the bead is defined by the area enclosed by the edge of the inner circle of the bead and the edge of its outer circle. For a bead with no defects, the width of the bead, defined by the distance between the inner and outer circle is minimal and somewhat constant. If at any point of the bead the wires are not held together by the cohesive force of the rubberized windings, this defect is visible as an increase in the width of the bead. Observing that any defect in the bead translates to a local change in its width, determines the detection strategy of the defects. The relevant features to detect a defect in a bead are embedded in the shape of the bead.

B. Acquisition moment and place

One of the requirements of the project is that the defect detection system should be implemented in the existing manufacturing process. Therefore, a moment and place should be identified where the defect detection can be added without disturbing the process. Requirements for an appropriate detection moment is that the bead should be available for individual inspection without occlusion. The bead apex machine is both the first and last moment in the manufacturing process in which a bead can be inspected individually without occlusion. An overview of the machine is given in figure 3. The model used in the manufacturing plant is the VMI APEXER.

By inspecting video material of the bead apex machine and on-site inspection of the machine, relevant operation characteristics are obtained from the machine. First, an individual bead is separated from a stack and placed onto the bead table. Here it is idle for at least four seconds, waiting to be transported by the bead setter to the drum, in which a rubber strip, apex, is applied to the bead. After the apex is applied to the bead, possible defects are not visible anymore because they are occluded by the apex material.

The observations that a non-occluded, individual bead is idle on the bead table for four seconds, makes it a perfect candidate for a detection location. An overview of the bead table is shown in figure 4.

The bead table consists of two plates, separated by a gap in which a gripper drags a bead onto the bead table until it touches the stopping poles. The center point of the bead on the table is determined by the outer radius of the bead and the position of the stopping poles. A disadvantage of the bead table as an inspection place is that the moving bead setter prevents that a camera can be placed directly above the bead table. Therefore, a top view of the bead cannot be acquired directly. To solve this, an acquisition system using two cameras is used to generate a top view from two side views.

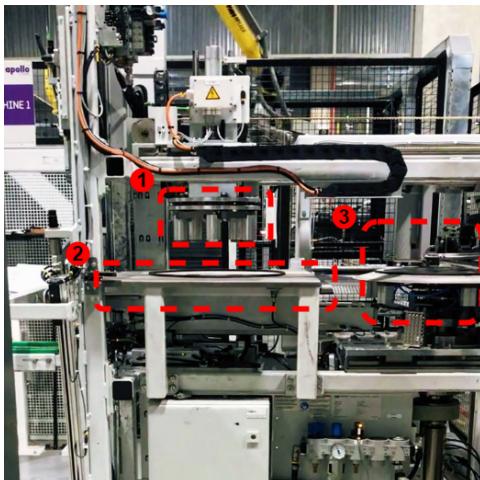


Fig. 3. The bead apex machine, with the components: 1) Bead setter, 2) Bead table, 3) Drum.

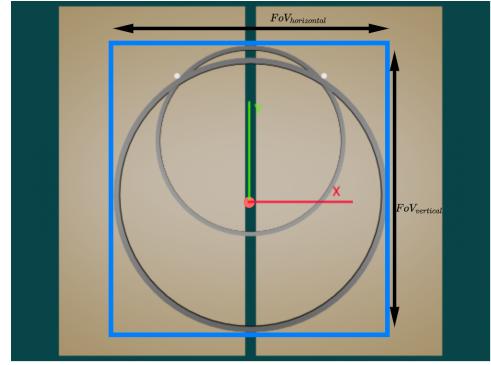


Fig. 4. Beadtable with the biggest and smallest bead, viewed from above.

TABLE I
SPECIFICATION TABLE

Specification	Value
What to detect	Deformations in bead shape
Acquisition location	Bead table @ bead apex machine
Required pose	Top view
Budget	€1000
Available time	4 seconds
Evaluation target	Detection accuracy
Minimal object size	1.1mm
Outer diameter bead	340-500mm
Effective object size (x×y)	500×525 mm

C. Object size

The effective object size is defined by the position of the smallest and biggest bead on the table. The center point of a bead on the table depends on the location of the stopping poles of the bead apex machine. The origin of the bead table coordinate system is placed in the middle of the bead table. The location of the stopping pole P_{pole} , on the positive x side of the coordinate system is $P_{pole}(x, y) = (135, 192.5)\text{mm}$. The radius of the stopping pole R_{pole} is $R_{pole} = 5\text{mm}$. The beads are placed in the middle of the table, which means that $C_{bead_x} = 0$. The y coordinate can be calculated using the Pythagoras theorem, illustrated in equation 2

$$C_{bead_y} = P_{pole_y} - \sqrt{(r_{bead} + r_{pin})^2 - P_{pole_x}^2} \quad (2)$$

Using equation 2, the location of the camera with respect to the table and the object size can be calculated as follows:

$$Y_{max} = C_{smallBead_y} + R_{min} = 251.15\text{mm} \quad (3)$$

$$Y_{min} = C_{bigBead_y} - R_{max} = -273.83\text{mm} \quad (4)$$

$$Size_Y = Y_{max} - Y_{min} = 524.98\text{mm} \quad (5)$$

$$Cam_y = Y_{max} - \frac{FOV_{vertical}}{2} = -11.34\text{mm} \quad (6)$$

Since $C_{bead_x} = 0$, the object size X, is determined by the diameter of the largest bead to be inspected which means that $Size_X = 500\text{mm}$.

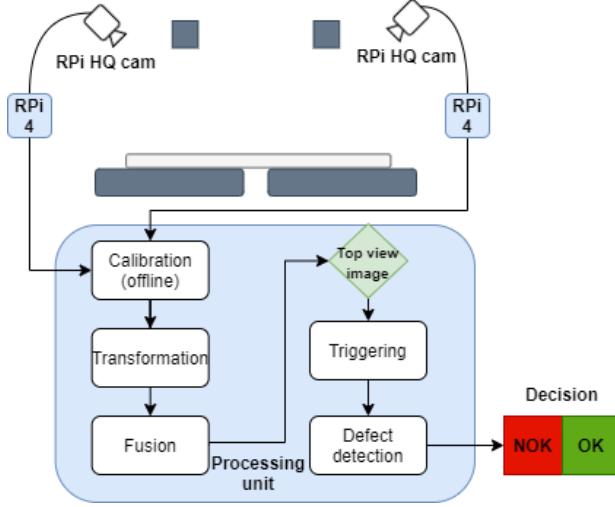


Fig. 5. The proposed system design of the defect detection system.

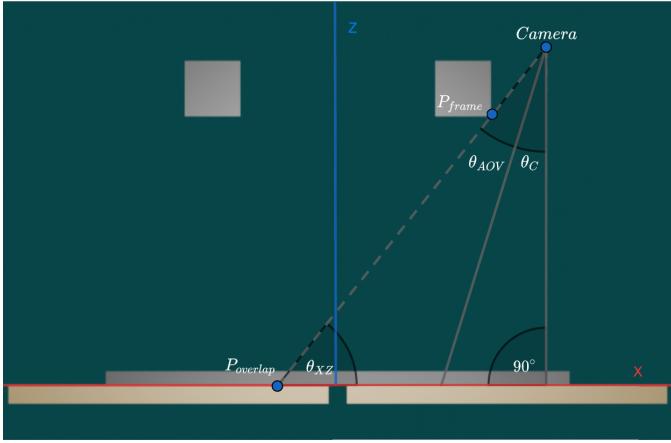


Fig. 6. Geometry of the camera calculation, shown in the XZ plane.

IV. DESIGN

Using the derived specifications from the previous section, a general system design is proposed, as shown in figure 5. The upper part of the figure represents the camera setup. Hardware selection and placement of the cameras is discussed in section V. The lower part of figure 5 represents the processing of the images which can be divided into two parts. First, the data coming from the two cameras is rectified and fused into one top view image, as described in section VI. Section VII describes the operations performed on the generated top view image to determine whether the image contains a bead with or without defects.

V. CAMERA SETUP

A. Camera location

Two cameras are used to construct a top view of the bead image. The cameras are placed such that their view is as close as possible to an actual top view. Figure 6 provides a geometrical overview of the situation. To calculate the required

camera positions, two points in space are important. The first point, $P_{overlap} = (x_{overlap}, z_{overlap})$, is located on the bead table and is placed just over the middle point of the bead table. This point is picked such that both camera's views have a small overlap which is required for fusing the two images. The second point, $P_{frame} = (x_{frame}, z_{frame})$, is the point located on the frame with the outer X coordinate. The line constructed by these two points, l_{xz} , defines the line of sight in which the camera can be placed which provides the desired view. The camera is placed on this line. The rotation of the camera with respect to the y-axis should be chosen such that the camera can just see the point $P_{overlap}$. This makes sure that the camera's view is as close as possible to the top view. This minimizes projection errors when the camera view is geometrically transformed. This can be observed in figure 6. Note that when θ_{xz} decreases, the difference in x coordinate between the point where the camera projection intersects with the top side of the bead and $P_{overlap}$ increases.

The rotation of the camera with respect to the y-axis, θ_{camera} is calculated using the angle of the line l_{xz} , and the vertical angle of view of the camera θ_{aov} as shown in equations 7-9.

$$\theta_c = 180^\circ - 90^\circ - \theta_{xz} - \theta_{aov} \quad (7)$$

with:

$$\theta_{xz} = \tan^{-1}\left(\frac{z_{overlap} - z_{frame}}{x_{overlap} - x_{frame}}\right) \quad (8)$$

$$\theta_{aov} = \tan^{-1}\left(\frac{h}{2f}\right) \quad (9)$$

where:

θ_c : is the angle of the camera with respect to the z axis

θ_{xz} : is the angle between x axis and l_{xz}

θ_{aov} : is the vertical angle of view of the camera

h : is the height of the camera sensor in mm

f : is the focal length of the lens

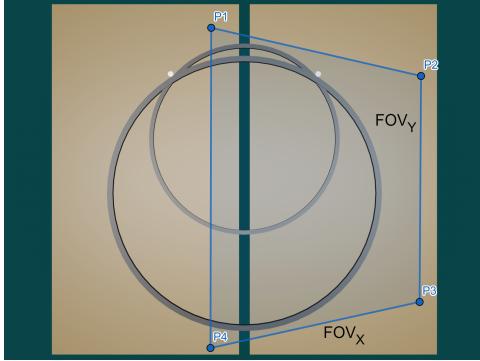


Fig. 7. Projected FOV of a camera viewing the object from an angle, shown in the XY plane.

The camera should be placed on the line l_{xz} such that the cameras field of view is at least equal to the effective object size in the y direction (see equation 5). The projected image plane on the bead table is shown in figure 7. The intersection points of the camera projection vector and the bead table can be calculated using the intersection of a line and a plane, as illustrated in equation 10.

$$\mathbf{p}_i = \mathbf{l}_0 + \mathbf{l} d \quad (10)$$

with:

$$d = \frac{(\mathbf{p}_0 - \mathbf{l}_0) \cdot \mathbf{n}}{\mathbf{l} \cdot \mathbf{n}} \quad (11)$$

$$\mathbf{l} = \begin{bmatrix} \cos \theta_c & 0 & \sin \theta_c \\ 0 & 1 & 0 \\ -\sin \theta_c & 0 & \cos \theta_c \end{bmatrix} \begin{bmatrix} w \\ h \\ 2f \end{bmatrix} \quad (12)$$

where:

- \mathbf{p}_i : is the intersection point of the camera projection vector on the bead table
- \mathbf{l}_0 : is the camera position (x_c, y_c, z_c)
- \mathbf{l} : is the (rotated) camera projection vector
- \mathbf{p}_0 : is a point on the bead table
- \mathbf{n} : is a normal vector to the bead table
- θ_c : is the angle of the camera around the y axis
- w : is the width of the camera sensor in mm
- h : is the height of the camera sensor in mm
- f : is the focal length of the lens

Using the above equations, the camera position can be calculated as a function of camera sensor size and focal length, which will be selected in the next section.

B. Camera hardware

The camera modules selected for this project are the Raspberry Pi HQ camera module [8], which is connected to two Raspberry Pi 4. The camera is the third camera module released by the Raspberry Pi foundation. Previous versions of the Raspberry Pi camera are used in a large variety of projects. The HQ cam is chosen because it features much flexibility on the hard- and software side. On the hardware

side, the camera module offers support for C and CS-mount lenses. These lenses are the standard in industrial vision, and therefore widely available. On the software side, the raspberry pi offers maximum flexibility due to its open-source platform. Support is widely available due to its big community, which allows for fast prototyping

The camera module has a Sony IMX477 CMOS sensor onboard with a diagonal of 1/2.3 inch. The sensors dimensions are $w \times h = 6.287 \times 4.712\text{mm}$, with 12.3 megapixels on it. Images are acquired with a rolling shutter and send to the Rasberry Pi by means of a serial data interface. The selection of lenses used in this project is mainly based on local availability. The lens used in this research has a focal length f of 6mm. The camera is configured using the picamera library [9], which offers a Python interface to the camera module. Camera settings such as resolution, frame rate, exposure time, and white balances can be set using this interface, along with color settings such as sharpness, brightness, and contrast.

C. Camera resolution

The required camera resolution can be calculated using equation 14. Three pixels are selected to be the minimal number of pixels per feature FP . The field of view observed by the camera follows from the projection of the image plane on the bead table, see figure 7. Using the specifications of the camera presented in the previous section and equations 7 - 12, the field of view can be calculated, along with the location and angle of the camera. An overview of these specifications is shown in table II.

$$R_s = \frac{R_F}{FP} \quad (13)$$

$$R_i = \frac{FOV}{R_s} \quad (14)$$

where:

R_i : is the required image resolution

R_s : is the spatial resolution (mm/pixel)

FP : is the required number of pixels per feature

R_f : is the size of the smallest feature to be detected (mm)

FOV : is the field of view observed by the camera

The resulting resolution of this analysis is 1670×1185 pixels. The camera's highest resolution mode is 4056×3040 pixels, but the camera is configured to capture images using 2x2 binning. Binning combines blocks of 2x2 pixels into one pixel. This process halves the resolution but increases the signal to noise ratio of every pixel. The resolution of the images captured by the camera is therefore 2028×1520 pixels

D. Interface and image format

Images captured by the camera system are sent to a PC for further processing. Both compressed and uncompressed images are available to the PC. Compressed images are acquired in the video mode of the camera system and compressed using JPEG compression. The images are transferred to the

TABLE II
PARAMETER TABLE

Parameter	Description Value
$P_{frame}(x, z)$	Position of the frame (mm)]
$P_{overlap}(x, z)$	Position of the overlap point (mm)
f	Focal length of the camera lens (mm)
w	Width of the camera sensor (mm)
h	Height of the camera sensor (mm)
$P_{camera}(x, y, z)$	Position of the camera (mm)
θ_c	Camera rotation around y axis (degrees)
$FoV(x, y))$	Field of view of the camera (mm)
R_{min}	Minimal required resolution (pixels)
	(165,300)
	(20,0)
	6
	6.287
	4.712
	(317.82, -11.34, 539.55)
	11.1
	(434.45, 612.64)
	1670 × 1184

processing unit by means of a HTTP web server. During experimentation, frame rates of 25 FPS are achieved. Raw images are available using the picamera interface in YUV420 format. YUV is a color scheme that encodes colors into an intensity channel (Y) and two color channels (U,V). The YUV420 format stores 4 bytes for 4 pixels for the Y channel. For the U and V channels, one byte per 4 pixels is stored. This means that 1.5 bytes per pixel are stored. Therefore, one image has a size of roughly 4.6 megabytes. The uncompressed images are available on demand by the processing unit, or as a continuous stream. When in continuous mode, experiments showed that frame rates of around 4 frames per second are achieved.

VI. TOP VIEW GENERATION

To fuse the images captured by the two cameras into one top view images, the images should be registered into a shared rectified reference frame. To do this, a calibration technique based on per-square transformations is proposed here. Figure 8 shows an overview of the calibration process. The main concept is to take an image of a checkerboard calibration pattern and make a mapping between the calibration pattern captured by the camera and a predefined rectified view of the calibration pattern. Geometric image transformations can then be estimated and used to transform images of the beads during online operation to a rectified view. Doing so for the two cameras, the images can be aligned using a marked middle point. The images are then fused into a single top view image.

Note that for every camera, only one image of the calibration is required to estimate the transformations, where traditional calibration techniques such as in [10] require images of the calibration pattern captured in different locations. The different operations involved in this method are now discussed.

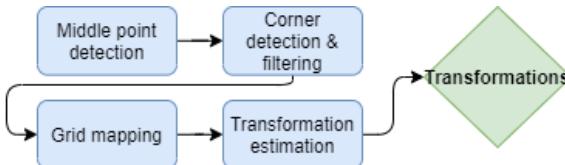


Fig. 8. Steps performed to estimate the transformations of the grid

TABLE III
PARAMETER TABLE FOR GRID MAPPING

Parameter	Value
H_{red}	$0 < H < 10$
S_{red}	$170 < S < 180$
V_{red}	$130 < V < 255$

A. Middle point detection

In the middle of the calibration pattern, a red marker point is placed. To detect this point, the image is converted to HSV color space which simplifies the color detection algorithm because it enables easier representation of the color. The pixel values which are in the color range shown in table III are marked as 1, the rest as 0. The resulting mask image is filtered for noise by means of morphologic erosion. Practical experimentation showed that 2 iterations of the structuring element cv.MORPH_ELLIPS of size 5 is sufficient to filter out the noise. Since the location of the middle point is only used to map it to the center square in the image, any non-zero pixel of the resulting mask can be used as the middle point and is picked at random.

B. Corner detection & filtering

Although software is available to detect calibration patterns in images, they are only capable of detecting rectangular patterns. Because the camera's orientation is not perpendicular with respect to the pattern, the shape of the pattern observed by the camera is not rectangular. Therefore, a custom calibration pattern detection algorithm is constructed. The corners of the calibration pattern are detected using the Shi-Tomasi corner detector [11] available in OpenCV as cv.goodFeaturesToTrack. The corners are refined using the OpenCV function cv.cornerSubPix. A critical parameter of the Shi-Tomasi corner detector is the quality level, which determines the minimal quality a corner should have. When this parameter is set too high, not all desired corners are detected. When too low, undesired corners are detected. To compensate for this, all corners are checked on their validity. For all points, it is checked if they are part of a valid square.

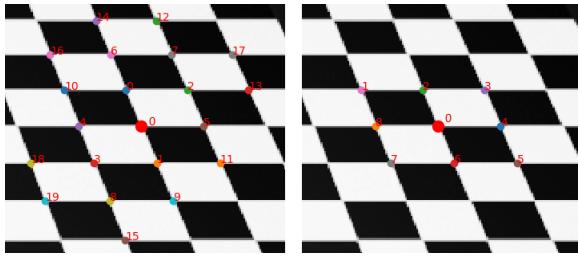


Fig. 9. Result of corner filtering. The left side shows selected corners before filtering, the right side after filtering.

To check if a corner is part of a valid circle, a 2×2 grid is fitted on every corner point, as shown in figure 9. First, $N_p = 20$ points with minimal Euclidean distance to the corner to inspect are selected. For these points, the angle with respect to the corner is calculated. P_4 is the point closest to the corner which has an angle $\theta_{margin} < \theta < \theta_{margin}$ where $\theta_{margin} = 10^\circ$. P_2 is selected as the point closest to the corner with angle $45^\circ < \theta < 135^\circ$. The angle θ_{02} between P_0 and P_2 is used to locate P_1 and P_3 . Opposite corners are used to find corners P_7 , P_6 , P_5 and P_4 .

Additional geometry checks are performed on the grid to validate that the shape of the grid is approximately a parallelogram. Points that are part of a valid square are marked as a valid corner point. Points which are in the middle of the 2×2 grid, P_0 in figure 9, are called valid query points which are used by the grid mapping algorithm.

C. Grid mapping

The main advantage of fitting the 2×2 grid to the corner points, it that a spatial relation between the valid corners is constructed. This relation allows the grid mapping algorithm to move through the corner points by means of up, down, left, and right. The grid mapping algorithm starts by selecting the query point closest to the middle point. From this point, it travels to the left top query point. This point is used as a starting point of the grid mapping.

Starting at the left top query point, the query point is moved down, until the end of the row is reached. Along the way down, it stores the upper left corner with respect to the query point. When the query point reaches the bottom of a row, it stores the left bottom square as well, and the query point is set to the point on the right of the query point on the top of the bottom. When the query point reaches the middle query point, the next time the query point is at the bottom, it will go back to the top query point. The query point moves down again and now stores the upper right square with respect to the query point. At the bottom of the row, it stores the lower right corner as well. This procedure is illustrated in figure 10.

The result of the grid mapping algorithm is that the dimension of the grid in terms of squares per row is known, along with the corner coordinates of these squares. These coordinates can now be used to estimate a geometric image transformation to a rectified square.

D. Transformation estimation

To transform the image into a rectified view, a geometric image transformation is calculated. The function `cv.estimateAffine2D` estimate the affine transformation coefficients between the source (X, Y) and target (x, y) points, see equation 15. As a target point, a square with PPS pixels in both horizontal and vertical directions is used, placed at the origin. When the transformations are known, the calibration procedure is complete, and the system is ready for normal operation.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad (15)$$

E. Transformation

The transformations are applied to each square of the image grid using the `cv.warpAffine` function. From the grid mapping procedure, the location of the square on the rectified grid is known and the transformed square is placed on the output grid accordingly. This results in a rectified image. Bilinear interpolation is used to determine the color information in the output image.

F. Fusion and cropping

When the images from both cameras are rectified, they should be aligned and fused into one image. The image of one camera is flipped upside down and mirrored, to ensure both images have the same orientation. The size of the output image is constructed, by calculating the maximum number of required squares in every direction counted from the middle square. The upper half of the output image is taken from camera 1, the lower half of the output from camera 2, resulting in a top view image.

VII. DEFECT DETECTION

As described in the case analysis section, the defects in the bead can be detected by measuring the width of the bead. The steps to obtain this measure and classify a bead either as a defect (NOK) or no defect (OK) are described now.

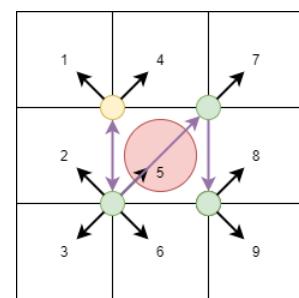


Fig. 10. Illustration of the grid mapping algorithm. The algorithm starts at the top left yellow query point.

A. Triggering

A trigger signal which tells the defect detection algorithm to analyze the image or not should be generated. The following situations can be identified in the input top view image.

- 1) No bead is visible
- 2) The bead is being dragged onto the bead table
- 3) Bead is idle on the table
- 4) Bead is picked up by the bead setter

When state 3 occurs, the trigger signal should be fired. Characteristics of this feature are that no movement has occurred in between the frames, and a circle is observed in the image. Movement is detected by calculating the absolute difference between consecutive frames. The resulting difference image is thresholded, and morphological eroded using the structuring element cv.MORPH_ELLIPSE of size 5. When the sum of the eroded image is zero, no movement is detected between the frames. When no movement is detected for $N_{movement}$ frames, the function cv.HoughCircles [7] is applied to the frame to detect circles in the image. If a circle of radius $R \geq R_{min}$ is detected, the trigger signal fires, and the bead is analyzed for defects using the method described in the next sections.

B. Segmentation

The first step of defect detection is segmentation of the bead. Background subtraction is the basis for this. An image of the scene is taken when no bead is placed on the bead table. This image is then subtracted from the sample image and the absolute value is taken from this difference, see equation 16. The difference image is then thresholded, such that only locations that contain changes between the fore- and background bigger than T_{diff} are present in the segmented image.

$$\begin{aligned} D &= |BG - I| \\ S &= D \geq T_{diff} \end{aligned} \quad (16)$$

where:

S : Thresholded difference image
 BG : Background image
 I : Image to be classified

Connected component analysis is performed to label the different regions present in the segmented image. Two pixels are part of the same region if their edges or corners touch (8-connectivity). If a region contains more than T_{area} pixels, the area is preserved in the segmented image.

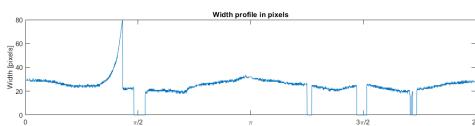


Fig. 11. Width profile of a bead containing a defect. The defect is visible as increase in the width of the bead. Regions with zero width indicate that no width information is available at that point



Fig. 12. Zoomed in version of a segmented faulty bead. Masking removes part of the bead, disturbing the width measurement

The final segmented image is obtained by binary masking the intermediate result. Pixels that are part of the binary mask are removed from the segmented image. Masking is performed to eliminate regions of the background where the segmentation with background subtraction is found to be not effective. These regions are the gap between the bead table, and the area around the screws and stopping poles, as shown in the right bottom image of figure 13. This condition is discussed in section IX. Figure 12 shows a zoomed-in version of a segmented image of a defect bead after masking.

C. Circle detection and inverse polar transformation

To simplify the width measurement, the representation of the bead is transformed from circular to rectangular using an inverse polar transformation. The center point and radius of the bead are extracted from the image using the Hough transformation. The implementation used here is the MATLAB function 'imfindcircles'. This implementation uses phase coding for estimating the radius of the circle. This method is described in detail in [6]. The radius and center point of the circle with the highest score is used to apply the inverse polar transformation.

The region of interest is defined as the area enclosed by two circles, with the same center point but a different radius. The inner and outer radii are derived from the radius found by the Hough transformation, with a subtracted (inner) and added (outer) margin of 10% of the image diagonal. The radii range and angle range are sampled and the corresponding image coordinates are obtained by using the relation between polar and Cartesian coordinates, see equation VII-C.

$$\begin{aligned} X &= R \cos(\theta) + cx \\ Y &= R \sin(\theta) + cy \end{aligned} \quad (17)$$

By specifying these coordinates in a mesh-grid format, the image can be resampled to a rectangular representation.

Nearest neighbor interpolation is used to find the gray values at the query points of the sampling.

D. Width measurement

Measuring the width becomes trivial when the image is transformed into a rectangular representation. The width is extracted by finding for every X value, the difference between the upper and lower white pixel. An example of a extracted width profile w is shown in figure 11. Locations with zero width are a result of masked out areas in the image.

E. Classification

Since defects lead to an increase in the width of the bead, defects can be detected by finding locations in the width profile w where it exceeds a threshold, see equation 18. The start and end locations of the defects are located by using the rising and falling edge of the error signal E .

$$E_i = \begin{cases} 1 & \text{if } w_i \geq T_{width} \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

The value of the threshold T_{width} is determined from a labeled database to tune the performance of the system. False-negative classifications should be prevented since this leads to a defect passing through the system, which is arguably less favorable than having a false positive classification.

VIII. EXPERIMENTS & RESULTS

To validate the performance of the system, the system is installed on the bead apex machine at the production plant located in Hungary. Additional lighting is installed on the machine to minimize the effect of varying illumination. The additional lighting changes the optimal camera position since the lights are placed such that the X coordinate of the point P_{frame} is increased. The lights are not installed symmetrically on the machine, and therefore the cameras are not placed symmetrically as well. No measurements of the physical dimensions of the machine after the lights are installed are available. Instead, the cameras are placed such that they are able to view the object as derived before, i.e. with θ_C in figure 6 as small as possible while having the required field of view. pps is set to 64 pixels per square. A calibration pattern is used with 15mm squares, such that 1 pixel corresponds with 0.234 mm. The camera system uses the uncompressed video stream, and camera settings are locked for acquiring images with constant brightness, color, and contrast. Triggering is configured to capture images when no movement is detected for $N_{movement} = 2$ frames, and a circle with a radius R_{min} which is equal to $\frac{1}{3}$ of the width of the sample image is present in the image.

The system is calibrated, and images of the background are captured. Then, normal production is started, with a bead of 15 inches in diameter, having 5 windings in both horizontal and vertical directions. Data of 1006 beads are collected. Furthermore, defect beads are placed on the bead table under various orientations. The defects in these beads are created by hand since no real defects occurred during operation. The total number of defect samples is 115.

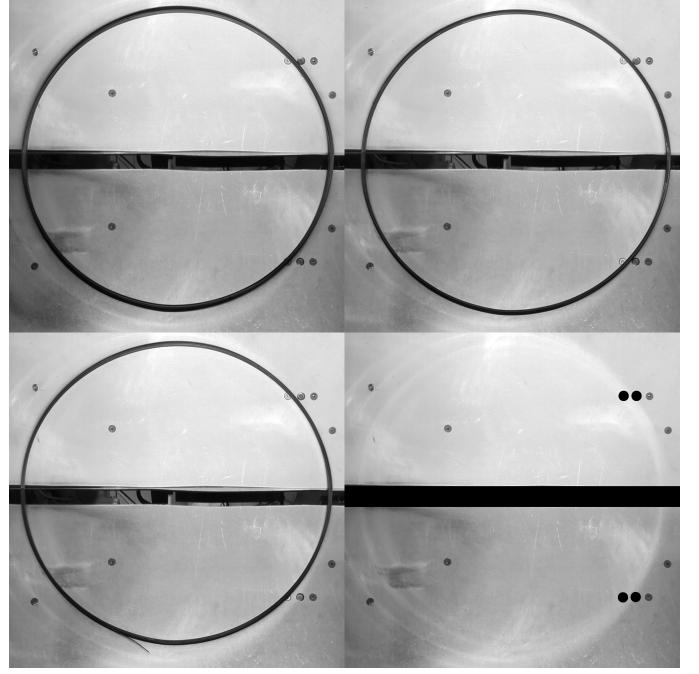


Fig. 13. 4 sample images collected in the data collection session. LT: Two beads stacked on top of each other RT: Image of a correct bead. LB: Bead with a loose wire. RB: Background image with masked regions

A. Results

Figure 13 shows 4 images captured using the acquisition system. The right bottom image shows the background image which is used for segmentation using background subtraction. Figure 14 shows the distribution of the maximum widths of both the correct and faulty beads extracted from the generated width profiles. One can observe in this figure that there is no overlap in distribution between the correct and faulty class. The smallest width of the faulty samples is 39 pixels and is shown in figure 12. The biggest width of the correct samples 37. This means, that by setting threshold T_{width} to 38, all samples of the dataset can be classified with an accuracy of 100%. The results will be discussed in the next section.

IX. DISCUSSION & FURTHER RESEARCH

The results presented in the previous section showed 100% accuracy on the collected dataset. Some remarks should be made about the results. The classification of the beads is based on the width profile which is extracted from the segmented region. However, some problems occur in the segmentation process. Due to the opening, closing, and moving of the bead gripper (figure 3), small vibrations occur on the machine. These vibrations can cause the camera to be in a slightly different position every time an image is captured by the camera system. This is not an issue when the background has uniform color, but big differences occur in areas with large contrast in the background image, resulting in false-positive segmented regions. Examples of these locations are along the edges of the screws and stopping poles on the bead table. A workaround for this issue is to mask out the areas where

X. CONCLUSION

Technical innovations are important to stay competitive in the tire industry. Defect detection systems can greatly reduce scrap material and therefore costs, by detecting the defects as early as possible in the manufacturing process. Currently, quality control of tires is performed by inspecting fully manufactured tires using x-ray images. This paper proposes the design of a machine-vision based defect detection that detect defects in tire bead as early as possible, within the existing manufacturing process.

The effect of tire bead defects is visible as geometrical changes in the shape of the bead, which can be detected by measuring its width. To measure this width, a top view image of the tire bead is acquired. The most suitable place to collect this data is to place the acquisition system on the bead table of the bead apex machine since the beads can be inspected individually without disturbing the existing manufacturing process. The dimensions of the bead apex machine don't provide room to place a camera directly above the bead table. This constraint, together with the limited financial budget leads to the design of a 2-camera acquisition system based on Raspberry Pi hardware. Analytical expressions are derived to calculate where the cameras should be placed to acquire the effective object size, and expressions are derived to calculate the required camera specifications. A novel calibration technique has been used to rectify the images produced by the cameras under an angle, which needs only a single image of a checkerboard pattern.

The width profile of the bead is extracted from the data produced by the acquisition system by means of background subtraction, thresholding, hough circle detection, and an inverse polar transformation. Classification of the width profile is performed by checking if the maximum measured width is above a threshold extracted from the data.

The system is installed on the bead apex machine on-site in Hungary to collect a dataset of 1006 correct beads and 115 faulty beads. The extracted width profiles allowed classification without errors, showing the potential of the system. However, some remarks are made about the method. Non-uniform background colors and local lack of contrast result in areas where defects can not be detected reliably due to segmentation problems. Further research is required to tackle these problems and to test the system with more and different bead sizes.

REFERENCES

- [1] R. Wang, Q. Guo, S. Lu, and C. Zhang, "Tire defect detection using fully convolutional network," *IEEE Access*, vol. 7, pp. 43 502–43 510, 2019.
- [2] Q. Zhu and X. Ai, "The defect detection algorithm for tire x-ray images based on deep learning," in *2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC)*. IEEE, 2018, pp. 138–142.
- [3] H. Shen, S. Li, D. Gu, and H. Chang, "Bearing defect inspection based on machine vision," *Measurement*, vol. 45, no. 4, pp. 719–733, 2012.
- [4] Y.-C. Chiou and W.-C. Li, "Flaw detection of cylindrical surfaces in pu-packing by using machine vision technique," *Measurement*, vol. 42, no. 7, pp. 989–1000, 2009.

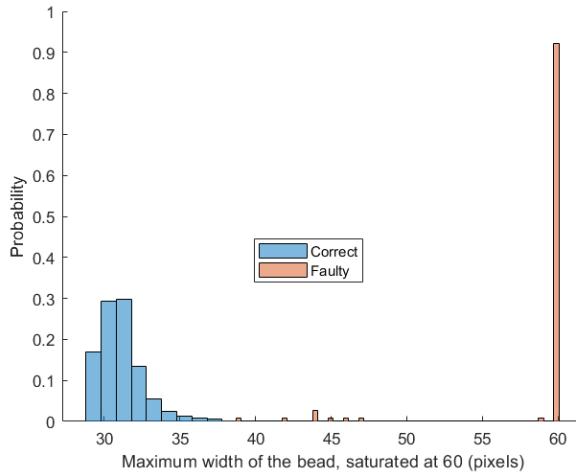


Fig. 14. Distribution the maximal extracted widths, values above 60 are saturated for visualization purposes

these problems might occur. The gap between the bead table is masked out as well since the contrast between gray values of the bead and the gray values of the background at that location are too low to be above the threshold T_{diff} , see figure 13. Masking out regions prevents unexpected segmentation results, but introduces blind spots in the defect detection algorithm because segmented defects might be masked out as well. To solve these problems, further research is required. Two strategies are suggested for solving these problems. The easiest one is to make changes to the physical situation by eliminating vibrations and altering the background to increase the contrast between the background and the bead, especially around the gap between the bead table. The second strategy involves research on alternative ways of segmentation, for example using edge information extracted from the scene instead or in combination with background segmentation. Also, a remark should be made about the feature used to classify whether the bead is classified as OK or NOK. Since the cameras are placed under an angle, the generated top view image contains geometrical distortions with respect to a true top view image. This can be seen as 'height shadows' around the generated top view image. This results in an elliptical shape of the bead instead of a circular shape, meaning that the width of a correct bead is non-uniform along its shape. This makes the classification less sensitive for defects at locations of the bead where the width of the bead is small when using the absolute width of the bead as a feature. On the contrary, the 'height shadows' in the top view image give some measure of the height of the bead. The classification is, therefore, able to detect the case that two beads are stacked on top of each other as shown in the left top of figure 13. Finally, more data should be collected of beads with different bead sizes to give a more clear overview of the true performance of the system.

- [5] D.-M. Tsai, "A machine vision approach for detecting and inspecting circular parts," *The International Journal of Advanced Manufacturing Technology*, vol. 15, no. 3, pp. 217–221, 1999.
- [6] T. J. Atherton and D. J. Kerbyson, "Size invariant circle detection," *Image and Vision computing*, vol. 17, no. 11, pp. 795–803, 1999.
- [7] H. Yuen, J. Princen, J. Illingworth, and J. Kittler, "Comparative study of hough transform methods for circle finding." *Image and vision computing*, vol. 8, no. 1, pp. 71–77, 1990.
- [8] "Raspberry pi high quality camera announcement," <https://www.raspberrypi.org/blog/new-product-raspberry-pi-high-quality-camera-on-sale-now-at-50/>, accessed: 24-06-2020.
- [9] "Pure python interface to the raspberry pi camera module," <https://picamera.readthedocs.io/en/release-1.13/>, accessed: 27-06-2020.
- [10] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [11] S. Jianbo and C. Tomasi, "Good features to track," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1994, pp. 593–600.