Implementation of timber volume control using photogrammetric technique

(Implementacja automatycznego pomiaru drewna korzystając z technik fotogrametrycznych)

Szymon Kosakowski

Praca inżynierska

Promotor: dr Jakub Kowalski

Uniwersytet Wrocławski
Wydział Matematyki i Informatyki
Instytut Informatyki

18 lutego 2022

Abstract

The problem of accurate log volume measurement is very important in the woodworking industry. Current technology of volume control according to national norms uses manual means, which turns out to be inaccurate and time-consuming. To overcome these inconveniences, an automatic method using image analysis is proposed. The proposed system was tested in practice in a sawmill.

Problem dokładnego pomiaru objętości drewna jest istotny w przemyśle drzewnym. Dotychczasowa technologia pomiaru, odnosząc się do obowiązujących w Polsce norm, opiera się na ręcznych pomiarach, które są niedokładne oraz czasochłonne. W celu zniwelowania tych niedogodności proponowana jest automatyczna metoda pomiaru, opierająca się na analizie zdjęć. Metoda została przetestowana w tartaku.

Contents

Chapter 1

Introduction

Although the problem of determining wood volume is essential in the woodworking industry, the level of measurement accuracy and automation is still not high. The most accurate method is measuring each log individually. Automated systems are provided for this method, but are rather expensive. Needless to say that measuring each log manually is too time consuming, especially for great amounts of wood. The most commonly used methods rely on average stack dimensions and coefficients which depends on wood class. Wood is classified in terms of its size, thickness, figure etc. The error of such measurements reaches up to 15%.

To automate measuring process and try to make it more reliable, photogrammetric technique is developed. Its main objectives were to speed up the whole process as well as eliminate the need for coefficient based on the wood class by trying to estimate the volume of each log individually.

1.1 Current Measuring Methods

According to polish national norms [4] the wood volume is measured as follows

- In case of wood on a truck:
  - width w is width of the cargo space,
  - length l is considered to be the nominal length of logs in the load,
  - height h is mean average of two measurements of load height.
- In case of wood in a stack(see Figure 1.1):
  - width w of a stack is measured in parallel to the lower plane of the stack on both sides of the stack. Resulting width is mean average of these measurements.

– length l of a stack is considered to be the nominal length of the stack measured with an accuracy of 1 centimeter.
– height h of a stack is measured every 1 metre and resulting height is mean average of these measurements.

Figure 1.1: Example height measurement of a wood stack
(image from [4])

Measurements are performed with accuracy of 1 centimeter. Resulting volume V is calculated using formula:

$$V = w \cdot h \cdot l \cdot x$$

Where x is a coefficient depending on wood class. Systematic error of these methods reaches 15%.

1.2 Existing approaches

Despite the fact that currently manual techniques are most commonly used, there were attempts to automate the process of timber volume measurement. A similar technique as proposed here was used in [7], where the proposed method relies on circle detection as well as 3D reconstruction of each log performed on a conveyor belt. The system proposed in [7] was tested in a laboratory on set of fifty artificial logs and showed accuracy level of 9%.

Similar system is also proposed in [3] which also uses circle detection algorithm and reference targets. For volume calculation stacking coefficients corresponding to ones in manual measuring methods are determined automatically. It is said that system [3] was tested in seaport conditions and provided fast and accurate measurements, surpassing existing manual techniques.

Chapter 2

Proposed System

This chapter describes in details how proposed system works. Input to the program is:

- front side photo of the woodpile
- distance from the camera to the other side of the woodpile D in m
- vertical distance between two reference targets *referenceTargetDist* in m.

The output is the summary volume of each log in the analyzed woodpile in m^3. The whole task was split into subproblems, each of them is described in sections below.

2.1 Image Acquisition

Images are obtained using ZED 2 Stereo Camera. The device is composed of stereo 2K cameras with dual 4MP RGB sensors. It has a field of view of 120° and is an UVC-compliant USB 3.0 camera backward compatible with USB 2.0. Left and right video frames are synchronized and streamed as a single uncompressed video frame in the side-by-side format. In this case it was decided to use the left lens as only a single image is needed. There is a SDK for c++ and python provided by Stereolabs where multiple parameters of the camera such as resolution, brightness, contrast, saturation can be adjusted during runtime and before initialization. Camera parameters are described in Table 2.1 along with its requirements.

In this case the 2K resolution was chosen as the system was tested to perform better on higher resolution images. The error of depth sensor was tested in [1] on 100 depth maps and Figure 2.1 shows the result.

Size	175×30×33 mm
Weight	159g
Image and depth resolution	HD2K: 2208 × 1242 (15 FPS) HD1080: 1920 × 1080 (30, 15 FPS) HD720: 1280 × 720 (60, 30, 15 FPS) WVGA: 672 × 376 (100, 60, 30, 15 FPS)
Depth	Range: 1-20 m Format: 32 bits Baseline: 120 mm
Lens	Field of View: horizontal 110°
Sensors	Format: 16:9 Pixel Size: 2-u pixels
Connectivity	USB 3.0 (5V / 380 mA) 0°C to +45°C
SDK System requirements	Windows 10, Ubuntu ≥ 16.04 CPU dual-core ≥ 2.4 GHz Minimum 4 GB RAM NVIDIA GPU ≥ 2GB Memory

Table 2.1: ZED 2 parameters description
source [5]

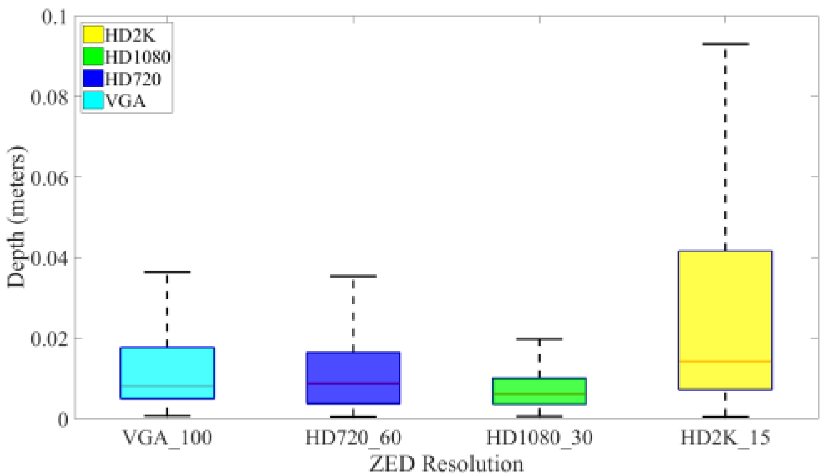

Figure 2.1: Box plot of standard deviations of depth ZED data
source [1]

2.1.1 Depth Calculation

Humans can have a three-dimensional perception of the world through the eyes due to the difference observed in the images formed in left and right retinas. In the imaging process, the images sent to the brain from each eye are not the same, with a slight difference in the position of the objects due to the separation between the eyes, which form a triangle with the scene points. Thanks to this difference, by triangulation the brain can determine the distance (depth) that the objects are in relation to the observer position. The implementation of stereo vision in computers uses this basic principle to recreate a 3D scene representation based on the two images of it taken from different viewing points. This is known as stereo reconstruction. In order to do stereo reconstruction, a series of steps are necessary, as calibration, rectification, and further depth determination.

The ZED camera computes depth information using triangulation (re-projection) from the geometric model of non-distorted rectified cameras, seen in Figure 2.1.1. Assuming that the two cameras are co-planar with parallel optical axes and same focal length $f_l = f_r$, the depth Z of each point L is calculated by Equation (2.1), here B is the baseline distance and $xi^l - xi^r$ is the disparity value . Notice that depth varies inversely proportional to the disparity.

$$Z = \frac{fB}{xi^l - xi^r} \tag{2.1}$$

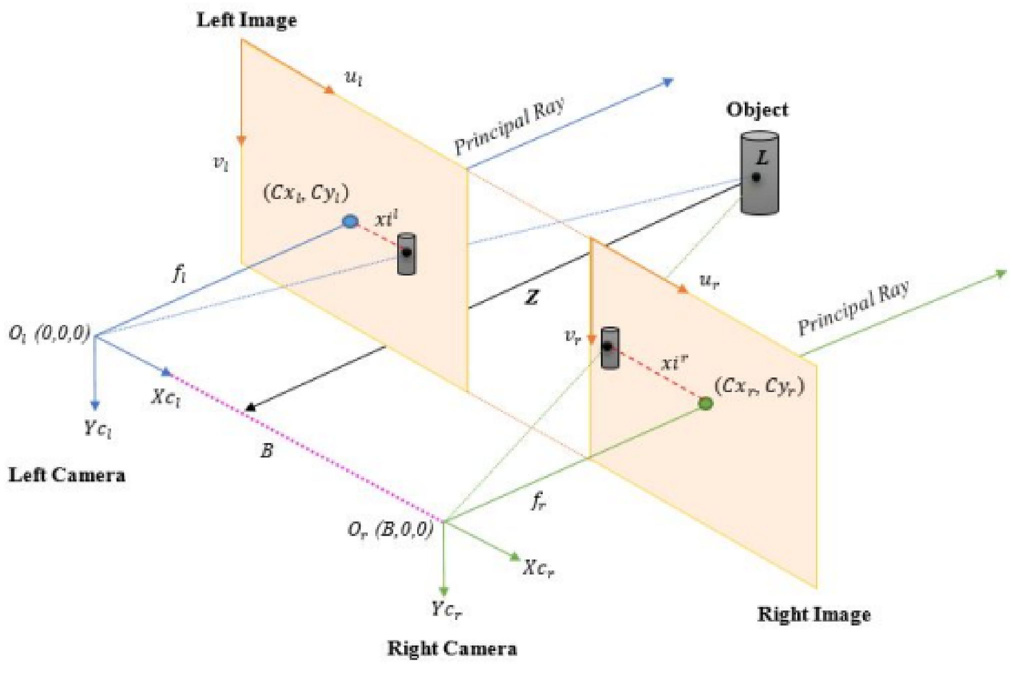

Figure 2.2: The operation principle of the ZED camera
image from [1]

2.2 Reference Targets

The size of each log end radius is to be found in pixels in the next steps.To be able to convert it to metres and calculate the log volume, reference targets are placed near the analyzed log pile. The vertical distance between two targets *referenceTargetDist* is measured manually before starting the program and is referenced during volume calculation. In total there are four targets placed near the pile in a way that all of wood which volume to be determined is included in the area described by the targets.

Figure 2.3: Original image

Figure 2.4: Image with targets found, centers of red circles are found centers of targets

If four targets are found, the program searches for log ends only in the area limited by them, which speeds up circle detection process significantly. If only two targets are found, the search of log ends is performed on whole image.

It was decided to use the reference targets because they are able to solve two problems at once: pixel to metre translation and optimisation. The search for reference target in an image is performed as follows:

1. Scale template image to *referenceSizeMax* size.
2. Detect edges in source and template image using canny operator.
3. Apply template image to every pixel in the source image.
4. If the template match value is bigger than some threshold, this place in source image is marked and its position saved.
5. Template image is scaled down by one pixel.
6. Process is repeated from step 3 until the template image reaches *referenceSizeMin* size.

Matching in step 4 is calculated using normalized cross correlation technique [2], where at point (x, y) of source image S the matching value $M[x, y]$ of template image T is calculated as follows:

$$M[x,y] = \frac{\Sigma_i \Sigma_j S[i,j] * T[i-x, j-y]}{\sqrt{\Sigma_i \Sigma_j S^2[i,j] * T^2[i-x, j-y]}}$$

Where i, j are positions in the area of S currently covered with T.

2.3 Log Ends Detection

To calculate the volume of wood in specified area, first the log ends are detected. It was assumed that log ends have circle shape in general. The process of finding circles in the image is performed as follows:

1. Find edges and their orientation in the image.
2. For every pixel in area of interest find the circle that fits best.
3. If found circle is decided to fit good enough – save its position and radius.

2.3.1 Preprocessing

To perform search for circles in given image some preprocessing operations are performed and additional data is saved.

First, the original image is converted to gray-scale and is smoothed with 3×3 Gaussian operator. The edge magnitude and orientation gradients are then calculated using Sobel operator:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * Img$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * Img$$

where * denotes a 2-dimensional convolution operation.

The gradients are then used to calculate angle of edges in whole image which are stored in *edge_angles*, where

$$edge\_angles(x, y) = atan2\Big(G_y(x, y), G_x(x, y)\Big)$$

The angle estimation accuracy is about 0.3 degrees according to [6]. When $G_x(x, y) = G_y(x, y) = 0$, the corresponding $edge\_angles(x, y)$ is set to 0.

Found edges are thinned to one pixel width with Canny operator and stored in *edges_thinned*.

2.3.2 Algorithm

As mentioned earlier the algorithm tries to fit a circle to every point of source image. The process of finding best circle for given point (x_c, y_c) in the image is performed as follows:

- First an area of size $max_r \times max_r$ with origin in (x_c, y_c) is selected from the the image, where max_r denotes maximum circle radius value. If the area would go out of bounds for given position (x_c, y_c) its size is reduced accordingly to image bounds.
- For each circle radius r in range $[min_r, max_r]$ starting from max_r a number of *valid* circle contour points is counted. Contour point p is counted as *valid* if both conditions are met:
  - The difference between angle of the edge p and angle of vector from p to circle origin $= (x_c, y_c)$ is less than *angleThreshold*,
  - Thinned edge was found in point p.
- Circle that was found to have the most *valid* contour points is then selected as best fit for position (x_c, y_c). Circle radius r_{best} and number of *valid* contour points of the circle n_{best} are then saved in matrices R, N respectively at position (x_c, y_c).

After the best circle for every position has been found, R and N matrices are analyzed and some circles are selected:

- Find position (x_{best}, y_{best}) at which ratio $circleRatio = \frac{N[(x_{best}, y_{best})]}{R[(x_{best}, y_{best})}$ is highest.
  - if $circleRatio > circleRatioThreshold$ save current circle, set values in area of this circle in N to 0 so another circle won't be found inside of current one and go back to step 1.
  - if $circleRatio \leq circleRatioThreshold$ finish the search.

2.4 Determining The Volume

In the final step of the algorithm, found circles are used to calculate the volume of analyzed wood. Radii of circles are given in pixels so at this step measured distance *referenceTargetDist* between reference targets is used to translate it to metres. Distance D from the camera to the back of analyzed wood pile is also measured before starting the program. It is assumed that logs are arranged in a way so they end in the same line on the other side of the pile. Figure 2.5 describes how the length of found log is determined.

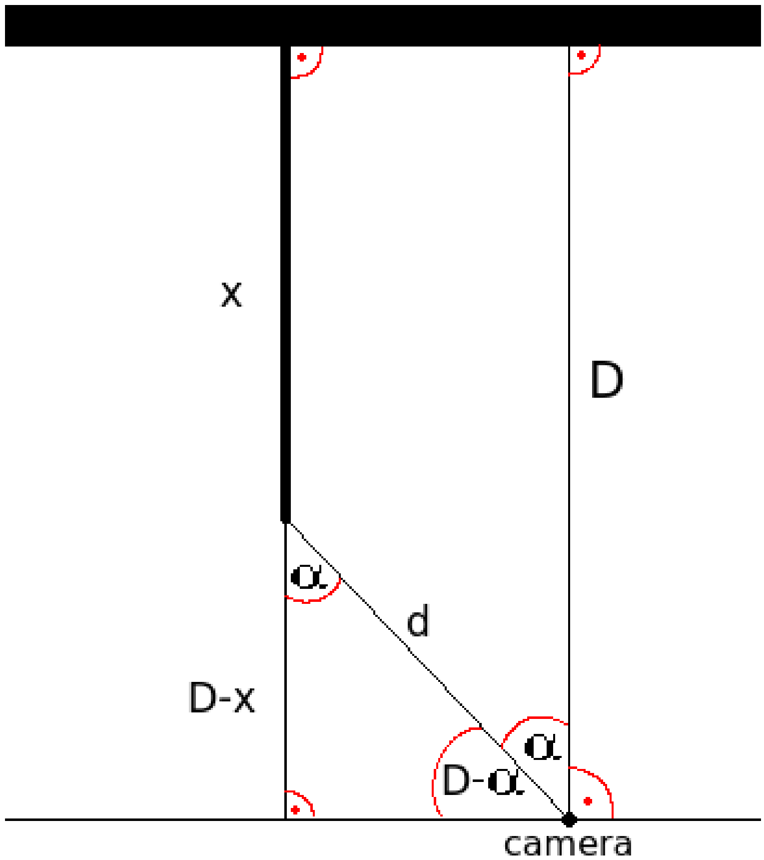

Figure 2.5: Projection of log finding on 2D plane

Where:

- d is distance of found log end to the camera
- α is horizontal angle between camera and found log, which can be estimated knowing resolution and field of view of the camera.
- x is wanted length of found log

So x can be found using law of sines:

$$\frac{D - x}{sin(90^\circ - \alpha)} = d \tag{2.2}$$

$$x = D - d * sin(90^\circ - \alpha) \tag{2.3}$$

After that, the volume of each log can be calculated $V_{log} = \pi r^2 \cdot x$ and volume of analyzed pile V is sum of volumes of all found logs $V = \Sigma_i V_{log_i}$.

Chapter 3

Testing

3.1 Parameters

During development of the program it was tested on few appropriate pictures of log stack and logs on truck chosen from internet. These pictures had similar but not the same resolution and pristine quality. Before first tests in the sawmill the parameters *circleRatioThreshold* and *angleThreshold* were determined using mentioned pictures. Circles of log ends were labeled manually and are visible in Figure 3.1.

Results of the circle finding algorithm were compared with labeled pictures using mean square error technique

$$\frac{1}{n}\Sigma_i^n err_i^2,$$

where

- $err = leftOut + nonOverlapping$
- Let *numLabeled* be the number of white pixels in labeled images and *numLabeledAndFound* the number of white pixels in logical *and* of found and labeled circles images. Then $leftOut = numLabeled - numLabeledAndFound$.
- *nonOverlapping* is number of pixels of found circles not overlapping with labeled ones.

Plot in Figure 3.2 shows result of these comparisons. MSE value was divided by 100000 for easier result reading.

Best values were found to be:

- $angleThreshold = 11°$
- $circleRatioThreshold = 0.8$

Figure 3.1: Example testing and labeled image

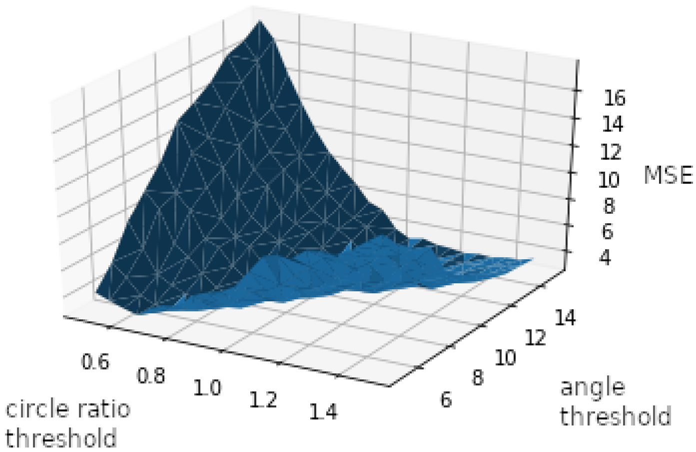

Figure 3.2: MSE value depending on parameters

3.2 Circle Detection Algorithm Testing

After the parameters have been determined the system was tested on images of appropriate logs found in nearby forest taken with ZED 2 camera. A short script in python was prepared allowing live preview of camera left lens image, modification of camera parameters during runtime and saving data from camera sensors.

Figure 3.3: Image of logs taken with ZED camera

No logs were found on above image. There was an attempt to adjust *angleThreshold* and *circleRatioThreshold*, but after loosening them the algorithm was founding circles in random places and not where the logs were positioned. The reason behind that is probably that there is not much contrast between background and the logs. Inside of the logs is also seem darker than previously tested images. This might cause noise in determining the angle of the edges as well as in finding edges themselves.

It was decided to perform the tests again during night and illuminate analyzed logs with flashlight.

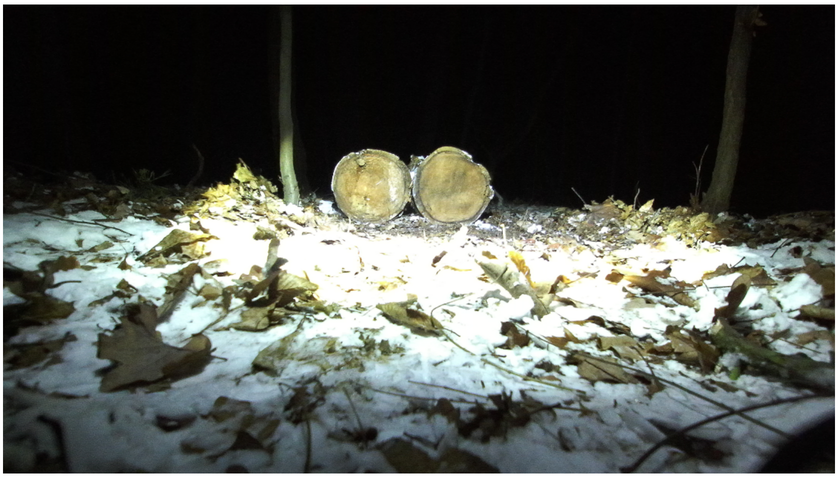

Figure 3.4: Image of logs taken with ZED camera during night

This approach seemed to remove suspected problems and the algorithm indeed properly found circles in above picture with $angleThreshold = 11$ and $circleRatioThreshold = 0.8$ values determined earlier.

Figure 3.5: Circles found on picture taken during the night

Correctness of distance sensor was also checked with these tests. The distance from left lens of the camera to both logs was measured manually on day and night attempt and compared with according read distances. The error in both cases was about 1.5%

Day vs Night Photos

Both photos were analyzed in terms of suspected problems. Below picture presents canny edge detection comparison as well as maps of angles of found edges on both pictures, day one the left and night on the right. Results are presented in Figures 3.6 and 3.7. Results were zoomed in on the area where logs were present so they would be easier to read.

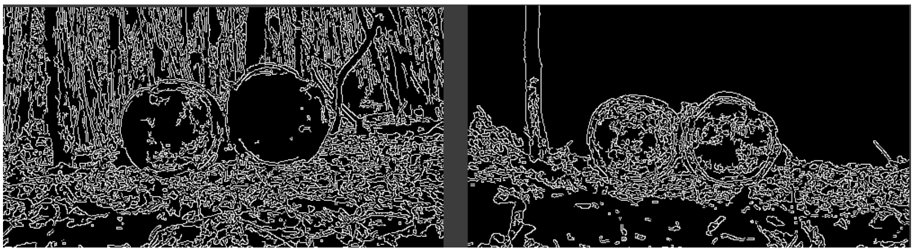

Figure 3.6: Comparison of Canny edge detection on day (left) and night (right) photos

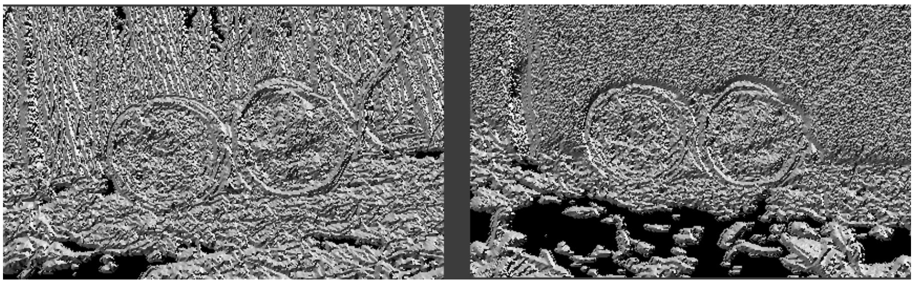

Figure 3.7: Comparison of Canny edge angle map on day (left) and night (right) photos

Comparison shows that there are more *log edge pixels* found in the night picture. Edges in the day picture are found mostly on the outside of the logs while in the night picture tangency of bark with wood is also detected which is helpful in circle finding algorithm as tested log ends are not expected to be perfect circles. It is also clearly visible that no background noise edges are detected in the night photo as only the first plane gets illuminated by the flashlight.

3.3 System Testing

In the next step, system was taken to the sawmill and tested as a whole. Due to previous observations tests were performed with sunlight and with artificial illumination.

Figure 3.8: Results of circle detection with sunlight

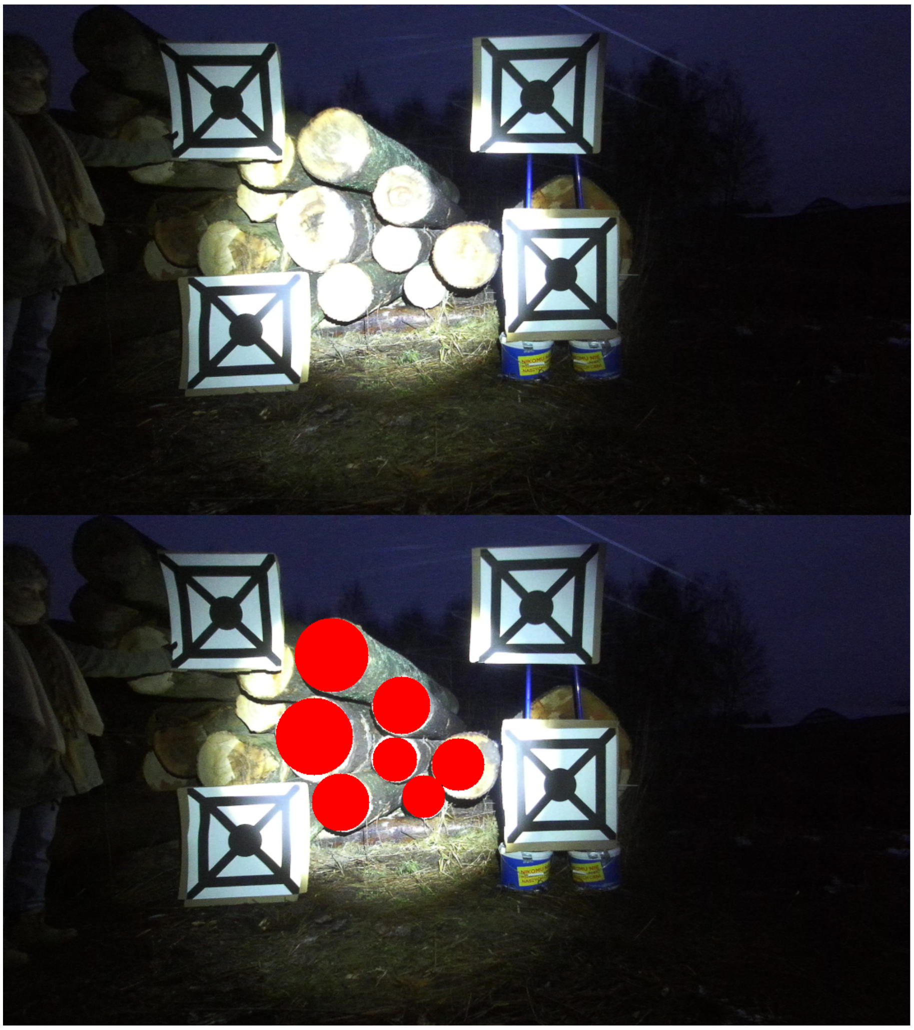

Figure 3.9: Results of circle detection with artificial illumination

Circle detection algorithm performed better with artificial illumination again, so this attempt was used to compare resulting volume.

Radii were converted to metres using found reference targets and compared against manually measured ones. The same was done with calculated log lengths. Below table presents error of these

	Radius	Area	Length
δ	7.16%	14.92%	14.14%

Table 3.1: mean relative error of radii, areas and lengths of found logs and their end areas

As described in previous section, there were already attempts to correct circle finding algorithm. Here an attempt to make log length determination more precise was made.

Current log determination algorithm assumes that every found log is on the same horizontal plane as the camera. This assumption imposes an unnecessary error so it was erased with taking 3D space into account.

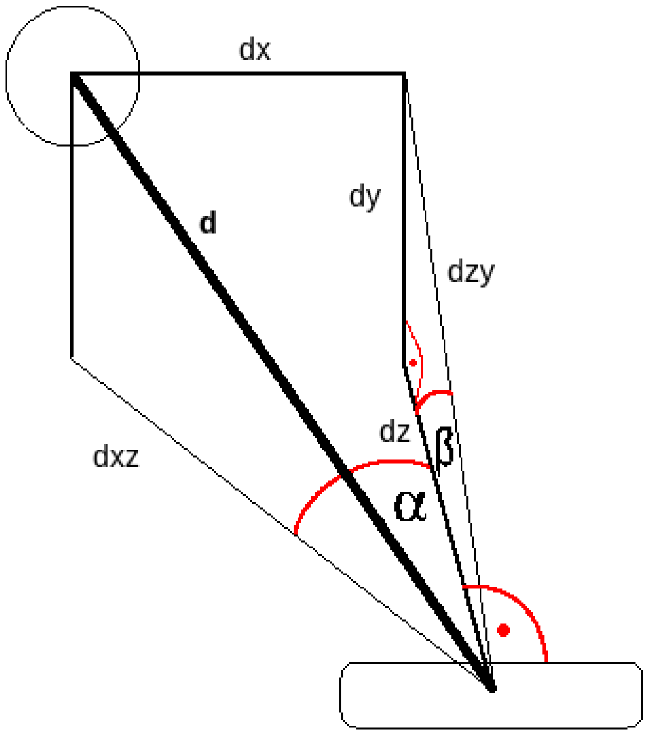

Figure 3.10: 3D length determination illustration

Where:

- α, β are horizontal and vertical angles respectively between camera and found circle center
- d_x, d_y, d_z are components on each axis of found distance d from camera lens to log center $d^2 = d_x^2 + d_y^2 + d_z^2$

With this, the new formula of calculating length x of found log is $x = D - d_z$. Using law of sines:

$$\begin{cases} \frac{d_y}{sin(\beta)} = d_{zy} \\ \frac{d_x}{sin(\alpha)} = d_{xz} \\ d_{zy}^2 + d_x^2 = d^2 \\ d_{xz}^2 + d_y^2 = d^2 \end{cases} \tag{3.1}$$

$$\begin{cases} d_y^2 = d^2 \frac{sin^2(\alpha)-1}{sin^2(\alpha)-\frac{1}{sin^2(\beta)}} \\ d_x^2 = d^2(1-\frac{sin^2(\alpha)-1}{sin^2(\alpha)sin^2(\beta)-1}) \end{cases} \tag{3.2}$$

So

$$d_z = d \cdot \sqrt{\frac{sin^2(\alpha)-1}{sin^2(\alpha)sin^2(\beta)-1} - \frac{sin^2(\alpha)-1}{sin^2(\alpha)-\frac{1}{sin^2(\beta)}}} \tag{3.3}$$

The relative error of the new method was $\delta = 20.18\%$. This is most likely because the logs didn't end on the same vertical plane as assumed and noise caused by constant distance D to the end of the stack is high, so even though calculations on camera end were more precise, it doesn't help. Error with previous method was smaller because calculated log lengths were too high due to inexact distance d_z.

Summary volume of analyzed logs was $5.12m^3$ measured manually. Calculated volume from found circles and lengths using 3D method was $6.66m^3$. Mean relative error was $\delta = 27.83\%$.

Chapter 4

Conclusions

An automated method for measuring woodpile volume was proposed. The volume is calculated with radii of found logs using circle detection algorithm and their lengths estimated using stereo camera. Error of proposed method $\delta = 27.83\%$ is higher than expected and results from inaccuracies of used solutions.

4.1 Reference Targets

The radii of found logs are converted from pixels to metres using vertical distance between two reference targets. This solution was chosen with assumption that analyzed logs end on the same vertical plane as positioned targets. In reality there are significant differences in position between logs and targets on this axis. Also it is not easy to correctly position the targets. Their size makes it unable to place them any further from camera than the log that sticks out the most in their area. Making smaller targets might be helpful but would not eliminate this obstacle. Also smaller targets might result in malfunction in detection process. It was observed that algorithm detects targets in random places more commonly when their size is smaller. This inaccuracy results in inexact values of log radii, leading to error in volume calculation.

4.2 Log Length Estimation

The lengths of found logs are estimated with similar assumption as with reference targets, that logs end on the same vertical plane on the back side of woodpile. With that, log lengths could be calculated as demonstrated in figures 2.5 and 3.10 with manually measured distance D from camera to the other side of woodpile. Similarly to problem described in previous section, the inaccuracy caused by this assumption is high. It is hard to estimate mean ending distance of analyzed stack and measuring many logs would miss the point of an automated system. A solution to this problem might be taking 2 pictures, one for front and back side of the woodpile with cameras at fixed distance. This way would also eliminate the need for manual measurements, but might add a need to suggest which log ends belong to the same log on both pictures. With this, error of determining lengths of logs would fall on accuracy of estimating distance and angle of given point in an image.

4.3 Comparison to Other Methods

Proposed system performed worse than mentioned methods. Obtained accuracy is three times worse than one in [7], but there the system was tested only in laboratory and used 3D reconstruction of each log so it is hard to compare these results. The system in [3] is said to surpass accuracy of manual method. Perhaps using coefficients inherited from manual methods is a good idea. System [3] was tested on huge timber stacks so it had to be hard to precisely analyze its results. Proposed system was tested on small amount of logs in order to be able to measure each log and compare obtained volume with real one. Maybe if it was tested on bigger part of the stack, the noise introduced by manual measurement of distance to the other side of stack D would not be as significant.

Bibliography

[1] Luis E. Ortiz Elizabeth V. Cabrera Luiz M. Goncalves. Depth data error modeling of the zed 3d vision sensor from stereolabs. *Electronic Letters on Computer Vision and Image Analysis*, April 2018.

[2] M. I. Khalil Ahmed Ibrahim. Quick techniques for template matching by normalized cross-correlation method. *British Journal of Mathematics & Computer Science 11(3): 1-9, 2015*, 2015.

[3] Vladimir A. Knyaz A. A. Maksimov. Photogrammetric technique for timber stack volume contol, August 2014.

[4] Zakład Informatyki Lasów Państwowych. Technical rules - regulations on wood raw material measurements. `http://drewno.zilp.lasy.gov.pl/drewno/Normy/2._zasady_przygotowania_do_pomiaru_-_ujednolicono_wg_zarz_54-2020_v3.pdf`, April 2019.

[5] Stereolabs. Zed 2 camera and sdk overview. `https://www.stereolabs.com/assets/datasheets/zed2-camera-datasheet.pdf`, November 2019.

[6] Opencv documentation. `https://docs.opencv.org/3.4/d2/de8/group__core__array.html`.

[7] Vladimir A. Knyaz Yury V. Visilter Sergey Yu. Zheltov. Photogrammetric techniques for measurements in woodworking industry, 2004.