



# Impact of tessellation surface model generation process on acoustical tire features

Michael Leupolz<sup>a,b,\*</sup>, Timotheus Radler<sup>c</sup>, Fank Gauterin<sup>b</sup>

<sup>a</sup> Mercedes-Benz AG, Stuttgart, 70372, Germany

<sup>b</sup> Institute of Vehicle System Technology, Karlsruhe Institute of Technology, Karlsruhe, 76131, Germany

<sup>c</sup> Daimler Truck AG, Leinfelden-Echterdingen, 70771, Germany

## ARTICLE INFO

### Keywords:

Photogrammetry  
Structure from motion  
Structured light  
Tread pattern  
Tire/road noise (TRN)  
Noise-related features

## ABSTRACT

Due to the electrification of vehicles, the focus of vehicle noise is shifting to the tires. Since tire noise is dependent of the tread pattern, methods to characterize tread patterns are required. This research investigates the effects of different processes of tire digitization on the identification of tread pattern features. An open source digitization pipeline is built to extract a tessellation surface model. This model is compared to one, generated with a commercial photogrammetric system. The comparison is based on three acoustically relevant tire features. Fully automated algorithms are introduced to extract these features from any 3D tire tessellation model. The research finds that although the resolution of the mesh surface of the proposed model is lower, feature recognition is not affected by this change. This paves the way for more accessible models that accelerate the statistical coupling of tire tread characteristics and their acoustic, handling or braking behavior.

## 1. Introduction

In 2012 more than 125 million people in the countries of the European Economic Area were exposed to higher noise directive indicators than the threshold of 55 dB  $L_{den}$ . As main reduction sources the authors mention, noise limits for vehicle exhaust and engine, quieter tires and low-noise road surfaces [1, p.40f].

Especially when noise threshold limits are exceeded, this can have serious impact on humans health. Even if sound pressure levels are not high but exposure is long enough, health problems such as sleep disturbances can occur [2]. This is why governmental institution regulate the overall emitted sound pressure levels of vehicles [3]. For the vehicle certification, the tire noise has a dominant role [4]. That impact only gets greater with the ongoing electrification of vehicles [5,6]. Additionally, in the subject of interior noise, tire noise is of grave importance. This is mainly due to it merely negative indications by the driver. While especially high powered combustion engines might be considered as pleasant sound, tire noise is always connected to negative emotions [7].

The tire tread is a complex structure that gives tire manufacturers the ability to modify a tire's wet and dry friction, and thus its braking and acceleration performance, vehicle stability, and sound emission.

Ejsmont et al. [8] conducted research on the impact of different tread features. The investigated tires were not mass-produced tires, but tires with hand-cut profile patterns that were acoustically measured

in an indoor road-wheel facility on two different drums. The authors determined the width and angle of transverse grooves, the existence of circumferential grooves, ventilation of pocket type grooves, and orientation of curvilinear grooves as important for the overall sound pressure level [8].

Stalter and Gauterin [9] investigated the impact of the circumferential tread pattern stiffness on sound pressure levels while applying torque on the tire. For that, two different tire tread patterns were hand-cut from smooth tires. The patterns were built with three tires each, each composed of different rubber mixture. The authors identified harder rubber, as well as tread patterns with longer blocks, to be louder. This is explained by analyzing the transferred tangential force in the contact patch. Thereby, high circumferential stiffness makes the tire slip over a longer distance in the contact patch, which causes a higher increase in noise level compared to noise generation mechanism without slipping [9].

Li et al. [10] compared the tire noise recorded in the near-field with a digital tire model. The tire tread was digitalized using a laser producing a highly accurate model. The recorded acoustical signal was divided into air pumping noise and tread pattern noise. These were compared to tire features in order to show correlations between the features and the acoustical signals. The method was applied to five different tires with similar size and aspect ratio [10]. The same authors later expanded the measured tires to a total number of 23, also with

\* Corresponding author at: Mercedes-Benz AG, Stuttgart, 70372, Germany.  
E-mail address: [michael.leupolz@mercedes-benz.com](mailto:michael.leupolz@mercedes-benz.com) (M. Leupolz).

similar size and aspect ratio. A difference of up to 17.2 dB(A) in the tread pattern noise was recorded. The acoustic signals were processed similar to the previously mentioned research and used to train an Artificial Neural Network (ANN). The ANN was able to predict the sound pressure levels for test and validation sets with an average error of 2.1 dB and 3.5 dB [11].

Oorath et al. [12] investigated different tire tread features and their impact on air pumping noise. For this, they developed an image scanning algorithm tracking air gaps entering or exiting the contact patch. The change in air volume in the tread as well as an assumed rubber compression are combined to a noise source, assuming a monopole source in the free field. From the representation as volume velocity source, a sound pressure level was then calculated. With the algorithm, tread pattern variations in groove width, contact patch length, groove length, groove angle and pitch offset in both shoulder regions were investigated [12].

Rapino et al. [13] investigated a similar approach to Li et al. [11]. 83 different tires were measured in a semi-anechoic chamber on a drum with sandpaper surface. Six microphones were placed in a semicircle around the tire in a distance of 1.5 m. Measurements were conducted at  $50 \text{ km h}^{-1}$ ,  $80 \text{ km h}^{-1}$  and  $110 \text{ km h}^{-1}$ . The acoustical measurements as well as 24 other parameters were used to train a statistical model. These parameters describe the outer shape of the tire (width, radius, and rubber hardness), the shape of the contact patch under axial load (width, length and roundness factor), the tread pattern (width of circumferential grooves and power spectrum in 14 1/3 octave bands) and operation conditions (rounding per minute and inflation pressure) of the tire. For that, the tire was digitalized with a laser profilometer. The trained ANN predicted the target output with a root mean square error of 2.3 dB(A). The trained model was particularly well suited to reconstruct the trend between tires, allowing the model to rank the tires by their sound intensity levels measured under the described conditions [13].

All mentioned research utilized either direct images of pressure profiles of the used tires or with laser scans. In general, there are different open source pipelines/reviews for image based 3D modeling of objects available [e.g. 14–16].

Following, we give insight into an open source pipeline built for the extraction of a tessellation surface model of a tire, as well as a commercial photogrammetry system. Different algorithms to extract the

- Air–Rubber ratio as area ratio (defined in Eq. (5))
- Circumferential grooves widths and locations
- Power spectrum of the tread pattern

are presented. According to recent research, these features are important for tire noise. The algorithms are then used to compare the tire models received through the different digitalization pipelines. Two tires are investigated, one winter and one summer tire. The herein proposed pipeline distinguishes itself from other, the authors known, open source solutions. This is especially seen in the *preprocessing* of the images where an image segmentation in relevant (tire) and irrelevant parts (rim and background) is done. This was shown to have a positive impact on the whole reconstruction process [17]. Furthermore, the *postprocessing* of the received model is highly adapted to the specific tire reconstruction purpose. This includes elimination of outliers in the mesh, the chosen meshing algorithm as well as the clustering of the final model which only selects the tire. These processes are further described in the following sections.

## 2. Reconstruction fundamentals

In this section, we give a definition of Structure-from-Motion (SfM), which is the underlying method to reconstruct our tires from image series. Furthermore, the steps of an incremental SfM pipeline are described. Finally, the densification of the reconstructed point cloud is discussed.

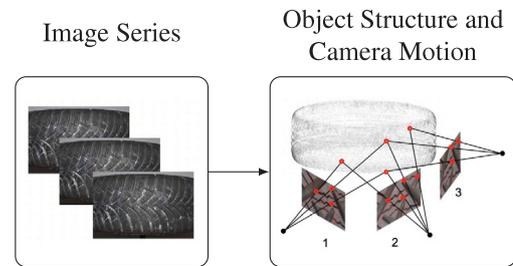


Fig. 1. Simultaneous determination of scene geometry and camera poses with SfM. Based on Özyeşil et al. [19, p. 307].

### 2.1. Structure-from-motion

The tire reconstruction is based on the SfM method. SfM can be assigned to photogrammetric methods and is used to determine three-dimensional structures from a series of overlapping images. The characteristic of the SfM method lies in the fact, that no precise knowledge of the orientations and positions of the camera, also referred to as pose, is required. Instead, the three-dimensional scene geometry and the camera poses are determined simultaneously, which is illustrated in Fig. 1. For this purpose, the extraction of corresponding feature points between the images in the series is used. These points are tracked across the image series and allow poses to be estimated, which can then be refined iteratively [18].

### 2.2. Incremental SfM-pipelines

SfM pipelines often follow an incremental structure and combine multiple steps to accomplish the reconstruction task, which is illustrated in Fig. 2. A SfM pipeline can be divided into two sub-steps, which are referred to as *correspondence search* and *incremental reconstruction* [20].

The *correspondence search* uses the image series as input and starts with the feature extraction, in which characteristic points are detected in each image. This is followed by a feature matching, in which feature points are compared across images and correspondences are matched. These corresponding points usually represent the same three-dimensional scene point. Nevertheless, outliers in form of wrong point assignments can still be present in the found correspondences. As a result, the *correspondence search* ends with a geometric verification. The latter describes the calculation of a geometric transformation between two images, which is intended to map a sufficient number of points from one image to the corresponding points in the other image. Correspondences, which do not hold the transformation, are detected as outliers. As a result, the *correspondence search* outputs geometrically verified image pairs in the form of a graph whose nodes represent the images and whose edges represent the associated image pairs. This is also called the scene graph [20].

The second step is the *incremental reconstruction*, which receives the previously determined scene graph as input variable. It starts with the reconstruction initialization, which describes the choice of an initial image pair. Usually, it is the pair with the largest point correspondence to provide a robust basis for the reconstruction. This reconstruction initialization is followed by an image registration, in which a new image is added to the reconstruction and in which the translation and rotation of the camera with respect to the world coordinate system are determined. It is estimated from correspondences between the already reconstructed three-dimensional scene points and the two-dimensional feature points contained in the newly added image. The image provides further point correspondences, which can be triangulated during the triangulation. Due to inaccuracies in the preceding steps such as the pose estimation, so called reprojection errors occur during triangulation [20]. This error describes the distance between a point detected

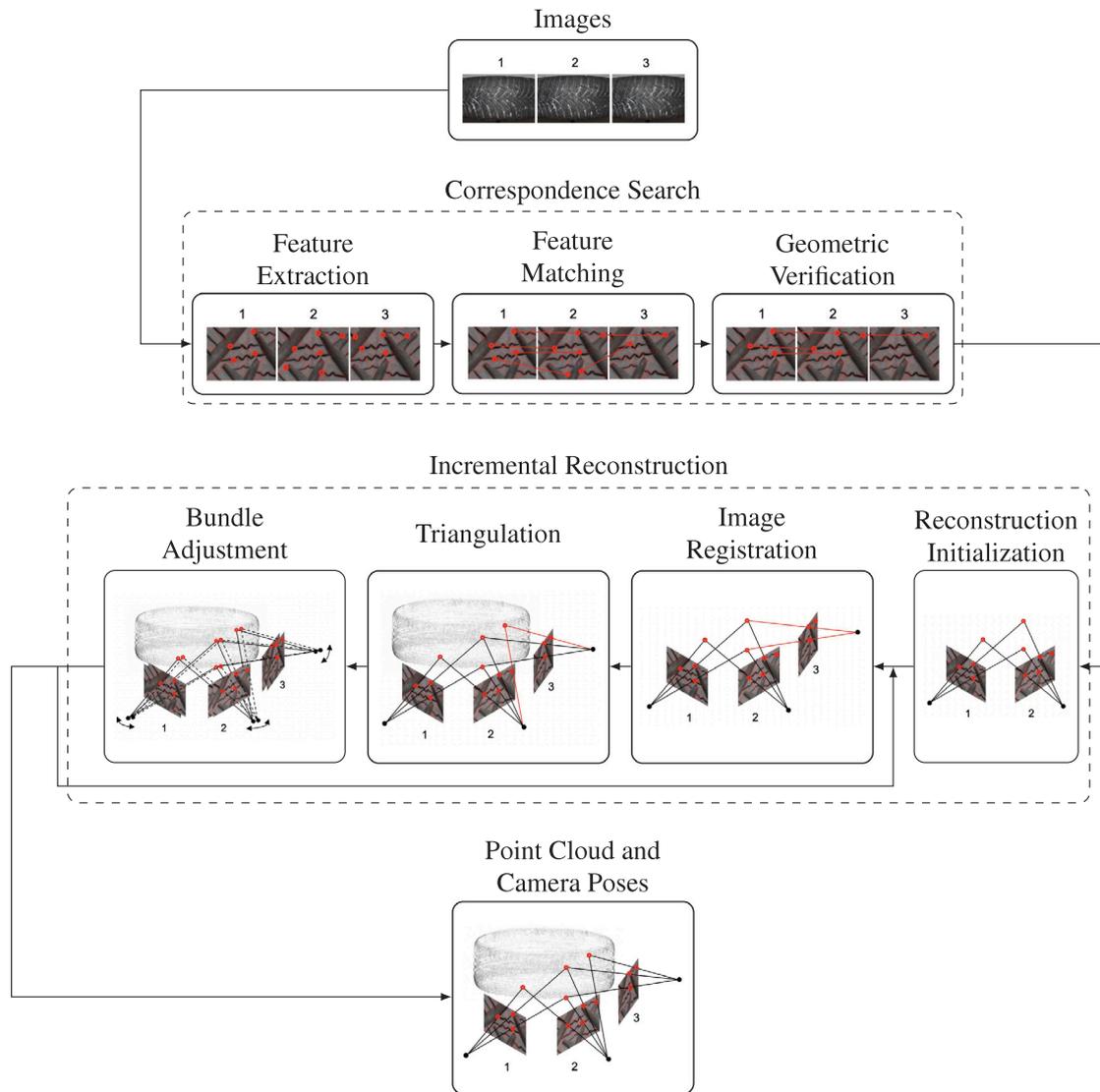


Fig. 2. Structure of incremental SfM pipelines. Based on Bianco et al. [20].

in the image and the three-dimensional scene point projected onto the image [21, p. 177–180].

Therefore, the last step of the *incremental reconstruction* is called bundle adjustment and tries to minimize the accumulation of these errors. Here, the positions of the reconstructed points and the parameters of the cameras are refined. The result of the *incremental reconstruction* is an estimation of the camera poses for each image and the reconstructed object as a point cloud [20].

### 2.3. Sparse and dense reconstruction

The point cloud created by the SfM method is sparse due to the small number of feature points, which are reconstructed three-dimensionally. To create a denser point cloud, the sparse reconstruction can be followed by a dense reconstruction. A pipeline, as shown in Fig. 3 can start with the calculation of depth maps. This calculation is done by estimating the depth of each image pixel. Two-View-Stereo algorithms construct a window around pixels and perform a *correspondence search* along the epipolar lines.

The latter are part of the epipolar geometry, which describes the geometric relationships between two views of the same scene [22, p. 14]. An important relationship is the epipolar constraint, which states that a point in the second image that corresponds to a point in the first

image must lie on a specific line, the epipolar line [23, p. 1332]. This reduces the search space for correspondences [24, p. 239–241].

Because many pixels can have similar appearances, Two-View-Stereo algorithms can lead to noisy depth maps. This problem can be countered by using Multi-View-Stereo (MVS) algorithms. Here, the epipolar geometry of all images that form an overlap is exploited and corresponding epipolar lines are determined to enable more robust similarity calculations for pixels. With the Multi-View-Fusion, the depth maps are then combined into a dense point cloud. In the last step, the surface reconstruction, the point cloud is transformed into a mesh [25, p. 36–38].

## 3. Reconstruction methods

This section describes our reconstruction pipeline for creating surface models from image series with its used methods and techniques.

### 3.1. Reconstruction pipeline

Our process of reconstructing vehicle tires from a series of images is a sequence of processing steps, which is why we refer to it as a pipeline. A first subdivision is the reduction of the pipeline into three steps called *preprocessing*, *processing*, and *postprocessing*, which is illustrated in Fig. 4.

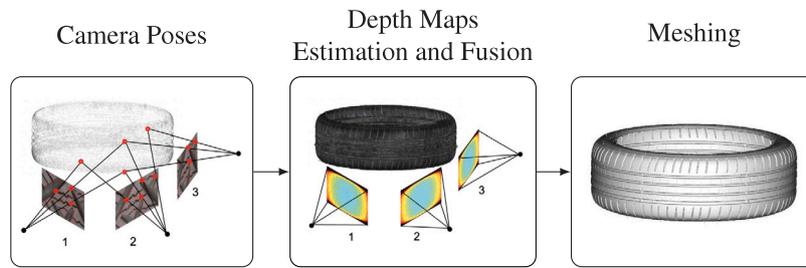


Fig. 3. Visualization of the dense reconstruction. Based on Schönberger [25, p. 36].

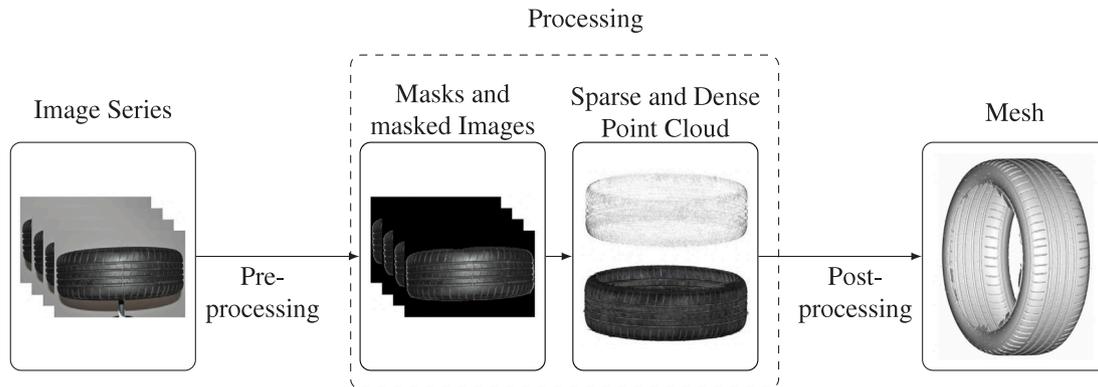


Fig. 4. Proposed reconstruction pipeline with a preprocessing, processing and postprocessing step.

The first step called *preprocessing* essentially describes a segmentation of each image, dividing the image content into relevant and irrelevant areas. This step is useful to digitize only significant image content, such as the tire itself, and to increase the quality of the final surface model. The next step is called *processing* and describes the reconstruction of the segmented images into a three-dimensional point cloud. It includes the estimation of the camera poses during sparse reconstruction and the generation of a dense point cloud of the reconstructed object during dense reconstruction. In the last step, the *postprocessing*, the quality of the point cloud is then improved by removing outliers and the cloud is transformed into a surface model with a suitable meshing algorithm.

The implementation of these steps is possible in various programming languages. Due to the existence of libraries for image processing, machine learning and the modification of point clouds and meshes, Python [26] is used as the scripting language.

### 3.2. Preprocessing

As mentioned before, the *preprocessing* step divides the content of each image in the series into relevant and irrelevant areas. The idea is to only digitize the relevant image area during the following steps. While the background and the rim are considered to be irrelevant parts of the image, the tire is a significant one.

The problem of filtering image areas is reduced to the creation of a binary mask, which is applied on the original image to form a masked image. Various methods are conceivable for generating the binary mask, for example thresholding, k-Means algorithm, or ANN. Own experiments showed that especially the latter one has proven to be particularly suitable. Our results are illustrated in Fig. 5.

The U-Net is used as ANN, which is supposed to deliver precise image segmentation with a small amount of training data [27]. The network is provided by the Python library *Segmentation Models*, which allows a simple and fast configuration of neural networks with pre-trained parameters [28]. Two separate networks are trained, one for background removal and one for rim removal. In combination, they allow the extraction of the tire.

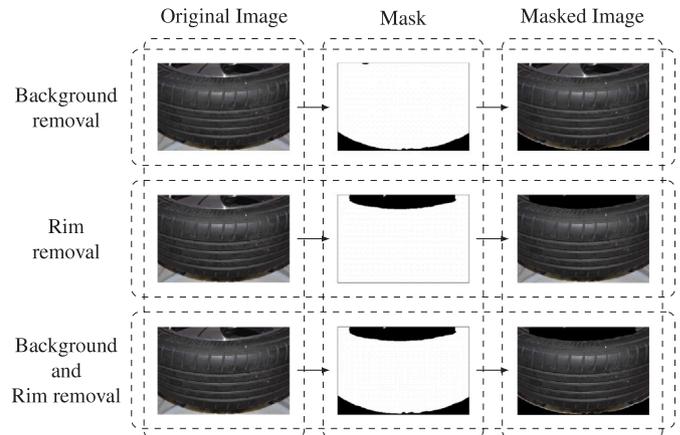


Fig. 5. Segmentation results with U-Net.

To train the networks, two datasets are generated. For this purpose, 100 images of different wheels are shot. These represented the wheels from various angles with different tires, backgrounds, and rims to ensure a high level of generalization capability of the trained networks. The images are then segmented manually with the Software VGG Image Annotator (VIA) [29], that allows the rim and background to be outlined with polygons. The outlined images are then exported as json-File and transformed via a Python script into corresponding binary masks. These images and masks are then scaled to a uniform size of 256x256 pixels due to the constant size input and output of the ANNs. Next, the two image datasets are enlarged with data augmentation. Existing images are transformed using techniques such as shearing, rotation or noise and added to the dataset. This is done with the Python library Albumentations [30] and has the purpose to prevent the networks from overfitting to few existing images and keeping their generalization capabilities high. Finally, two datasets for rim and background detection

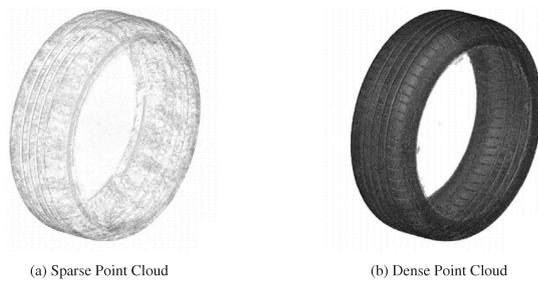


Fig. 6. Sparse reconstruction with OpenMVG and dense reconstruction with OpenMVS.

and removal were available, each of which consists of 1500 images including the corresponding masks.

For the training, the previously created image datasets are divided into training data consisting of 1200 images and test data consisting of 300 images. The datasets are propagated through the networks 100 times with a batch size of 8. The Dice Loss function delivered good results. Fig. 5 shows well performing masks to separate rim and tire.

### 3.3. Processing

The *processing* step describes the reconstruction of the tire as a point cloud from the previously segmented images. SfM is used as the underlying method for estimating camera poses and the sparse point cloud, while MVS algorithms are used to create the dense point cloud. Because of the high complexity and the high effort involved in implementing these algorithms, existing software packages are used. There are multiple open source solutions such as Meshroom [31], COLMAP [32,33] or OpenMVG [34] in combination with OpenMVS [35].

We used the latter two packages for the tire reconstruction. OpenMVG and OpenMVS come with precompiled binaries for e.g. feature computation, feature matching, camera pose estimation and sparse reconstruction as well as point cloud densification and meshing, which together form a complete SfM and MVS toolchain. This toolchain is purely console-based and therefore its executables can be easily used and started within a Python script. Furthermore, initial tests showed, that the calculations of OpenMVG v1.6 and OpenMVS v1.1.1 run purely on the CPU and do not require a dedicated graphics card. Due to that, the reconstruction can be done on any computer without portability issues. OpenMVG v1.6 comes with one global and two incremental SfM methods for camera pose estimation and sparse reconstruction. We used the first incremental method called IncrementalSfM. Fig. 6 shows the difference between sparse and dense point cloud.

### 3.4. Postprocessing

The *postprocessing* steps essentially describe the removal of outliers in the reconstructed point cloud, the meshing of the point cloud, and the eventual smoothing of the mesh. The focus is on choosing a suitable meshing algorithm. There are multiple different meshing methods that can be used. These are, for example, the Ball-Pivoting Algorithm (BPA) [36] or the Poisson Algorithm [37]. Both algorithms are provided by the Python library Open3D [38].

The BPA is based on the idea that a triple of points should form a triangular plane if they are touched by a ball of defined radius. This ball then pivots around all edges of the newly generated triangle and the collision with another point is checked. If a collision occurs, the edge and the point are combined to form another triangle [36].

The Poisson Algorithm on the other hand, is based on calculating an indicator function, which is zero outside and one inside the model. This leads to the fact that its slope is non-zero only near the surface. There it corresponds to the inward surface normal, so a cloud with oriented points essentially samples the gradient. The name of the

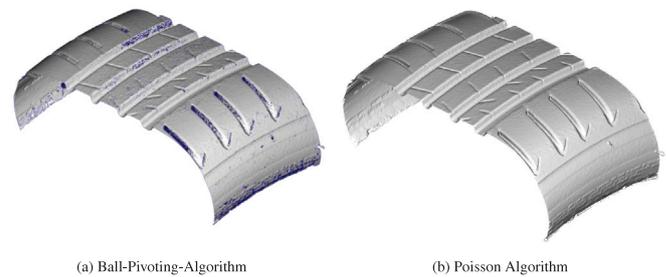


Fig. 7. Meshing results with Ball-Pivoting Algorithm and Poisson Algorithm. Holes in Mesh are marked in blue color.

algorithm results from the possibility of formulating the search for an indicator function, whose gradient approximates the sampling points, as Poisson equation. If the function is known, it implicitly specifies the reconstructed surface, which can then be extracted in further steps [37].

To mesh our point cloud, the Poisson Algorithm is preferred over the BPA, because it produces watertight meshes [37]. For the BPA, there is a risk that it can deliver meshes with holes, which is illustrated in Fig. 7. Furthermore, tires can have a very fine profile structure, which is why a very small ball radius may have to be selected for the BPA to be able to resolve these structures well. In combination with large tread blocks and the relatively flat and structureless sidewalls, longer calculation times are observed with BPA.

## 4. Structured light

As an alternative to the previously explained reconstruction pipeline, a commercial system using a structured light 3D modeling approach, is used. Following, we refer to the commercial model as reference model.

The structured light approach needs at least one camera and one projector [39]. The relative position between the cameras and the projector needs to be known [40, p. 238]. The projector then projects a light pattern onto the object. For the construction of the light pattern, different approaches are possible. The light pattern is referred to as codification, since it allows to uniquely identify each surface point of the object [39]. For example, a binary structured codification is achieved through projecting different binary intensity patterns with 0 (black) and 1 (white) onto the object. By varying the orientation and thickness of each line of the pattern, a unique codeword for each pixel is created. Using the codewords, camera, and projector position, the 3D coordinate of each pixel is calculated [39].

The used sensor *ATOS III Triple Scan* from the former company GOM GmbH nowadays Carl Zeiss IQS Deutschland GmbH works with two camera systems. The object is thus viewed from different angles each time a picture is taken. This allows the visualization of deeper pouches and the process gets more stable since multiple pictures overlap, making it possible to calculate the coordinates of some points multiple times [41,42].

Furthermore, the use of two camera systems allows a correction of thermal expansion in the camera system through the use of both cameras as triangulation basis. This allows a wider use of the system without the need for recalibration [43, p. 50].

The system used in this research does not apply a single binary black and white projection but uses multiple different codes with phase difference and blue light. Blue light makes it easier to detect objects under bad lighting as well as shiny ones. As mathematical basis for the triangulation, the distance between the two cameras is used [41,42]. The system outputs a fully meshed tessellation surface model.

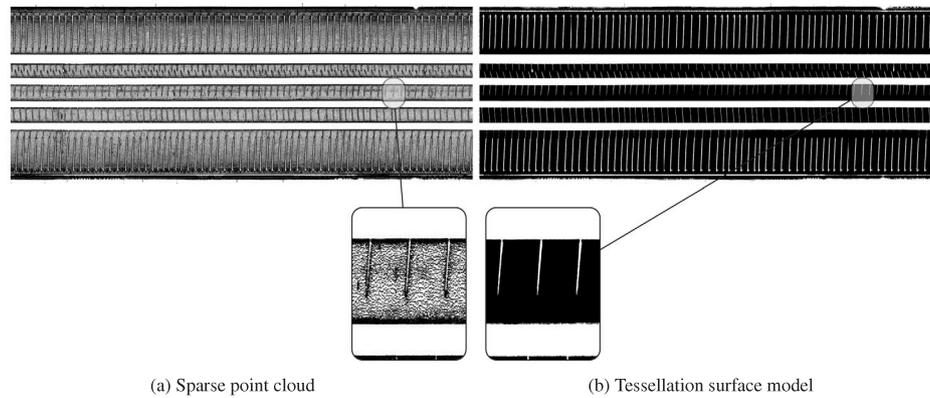


Fig. 8. Tread pattern images with grooves being cut out for a summer tire as sparse point cloud and connected tessellation surface model.

## 5. Image processing

In Section 3 we described the process of reconstructing a 3D tire model from images with either a self constructed pipeline that only uses open source packages which was developed in a Master’s Thesis [17] or a commercial software. Even though some tire features might be extracted directly out of these models, some are more easily accessible from a standardized tread image or require a combination of both.

### 5.1. Extraction of standard images

To gain more insight in tire tread parameters as e.g. Air–Rubber ratio, number and width of circumferential grooves as well as quantity of grooves, we create a standardized digital footprint of the tire tread that resembles the imprint of the tread pattern that results from the tire rolling on the ground without axial load applied.

Fig. 8(a) shows such a resembling tread pattern image with only the grids being shown. Both of the previously explained algorithms to create a 3D tire model do not only create a sparse point cloud but fully meshed tessellation surface model. A set of three grid points (nodes) is connected to a triangle. Each node can be referred to with a number as well as three coordinates  $x_1$ ,  $x_2$  and  $x_3$ . To each triangle a specific number is assigned. Each of the triangles also refers to the corresponding three nodes. The resulting connected surface is shown in 8(b).

The nodes and triangles do not have any specific numbering rule. Due to this, to extract all nodes that do not lay on the upper surface of the tire, the nodes are first sorted according to their  $x_2$  (for coordinate system refer to Fig. 9) coordinate.

After sorting the nodes, they are divided into chunks of the width 0.5 mm by their  $x_2$  coordinate. This creates circumferential lines of the specified bandwidth. These are necessary to extract the real circumference for each  $x_2$  coordinate. One of these lines is presented in Fig. 9.

Next, the distance between the tire axle and each node is calculated through

$$d = \sqrt{x_1^2 + x_3^2} \quad (1)$$

With the point distances and the categorization into axial chunks, a tire reference radius is defined. For the reference radius, the 95 quantile of all points in each chunk is calculated. This ensures that no single grid point, which might result from measurement errors, defines the radius of a chunk. The defined radius is then compared to the distance of each point in the specified chunk. Every point that does not meet the criteria

$$r - \Delta r \leq d \leq r + \Delta r \quad (2)$$



Fig. 9. Tessellation surface model of tire and rim with marked tire width chunk; chunk amplified for visibility.

with  $r$  as previously calculated radius and  $\Delta r$  as user defined threshold, is deleted. Therefore,  $\Delta r$  specifies the depth of grooves that remain in the model. The upper bound is kept to eliminate possible outliers. The calculation of a flexible reference radius  $r$  is necessary for two reasons. First, the tire default radius calculated through the rubber width, aspect ratio, and wheel radius is impacted by production tolerance. Second, using the default radius of the tire, the whole shoulder part would be lost due to the curvature of the tire.

Nevertheless, the default radius is still important. This is especially the case for tires with circumferential grooves. In these cases, it is possible that all grid points of one chunk lay in a circumferential groove. This results in the groove being defined as  $r$  so that almost no grid points fail Eq. (2). To avoid these cases an additional criterion

$$r_{\text{default}} - r < r_{\text{threshold}} \quad (3)$$

is defined in cases, that do not satisfy Eq. (3), and are situated in the center region of the tire,  $r$  is set to be  $r_{\text{default}} \cdot r_{\text{threshold}}$  needs to be larger than the maximum depth of the circumferential grooves.

The center region of the tire is determined by four steps, presented in Fig. 10. First, all grid points of the tire are projected on one surface with every point being way below the default radius being neglected for clarity reasons. This leads to a scatterplot that shows a bandwidth of possible radii for each width position on the tire. Also, this method avoids that by taking only one cross section, the different regions might be altered by local grooves. Second, the silhouette of the scatterplot is extracted by comparing every point in a given  $x_2$ -chunk and only keeping the largest one in the specific chunk. This way the second image is received. Third, a Savgol filter [44] is applied. This is done, to eliminate single outlier, which would alter the last step. Lastly, a

central discrete differentiation, according to

$$\frac{\delta r^i}{\delta x_2} = \frac{r^{i+1} - r^{i-1}}{x_2^{i+1} - x_2^{i-1}} \quad (4)$$

is calculated [45]. This allows to look for the smallest  $x_2$  value with a slope that is smaller or equal than zero and the last  $x_2$  value with a slope that is greater or equal to zero. These points identify either the point where the first circumferential groove starts, respectively the last one ends or the first point, respectively the last point forming a flat rubber surface area and therefore define the center region of the tread pattern.

Every grid point is then, according to his position on the tire, either compared to the default radius or its individually calculated radius as stated in Eq. (2). For each triangle, it is checked whether all referring grid points are kept in the model, otherwise, they are eliminated from the model as well. This procedure leads to a tessellation surface model of the tire that does only include elements on the upper surface of the tire. To generate the images shown in Fig. 8, the remaining grid points are transformed via polar coordinate transformation in a manner that the tire surface is virtually rolled out. The relation between every grid and according triangle remains the same, which allows for the plot to include the triangles and not only the grid points. Resulting in not only a scatter plot but a fully closed surface plot. The coordinate transformation is achieved through cutting the tire in half, transforming each half and then adding the flattened parts back together. This is necessary because of the periodicity of the transformation. As seen in the filtered silhouette curve, usage of the largest value in each chunk, might lead to outliers defining the surface. Still this is considered a better approach than the usage of a e.g. median values since this leads to a loss in the precision of the first groove, which especially occurs in tires with slanted block edges as seen in Fig. 14(b).

## 5.2. Feature extraction

To compare the resulting standardized images as well as the tessellation surface model themselves, some tread features relevant to tire noise are calculated. First, the Air–Rubber ratio which basically defines how much of the rubber will be in contact with the road is calculated. Thereby, a smooth road surface is assumed and the effect of the axle load on the tire is neglected, which would lead to deformation and a change in the groove shape. Second, the number and width of circumferential grooves. Third, the average length as well as frequency of different lengths in the grooves is investigated.

### 5.2.1. Air–rubber ratio

The Air–Rubber ratio basically shrinks down the whole tire tread to one number. It can be calculated from the standard images extracted earlier. Nevertheless, these images basically show the whole tire with some parts of the sidewall, which makes it necessary for a valid comparison to focus only on the tread that is in contact with the road in reality. Even though the Air–Rubber ratio allows some insight in the tires constitution, it neglects orientation and shape of each groove/block itself. To consider these, the shape of a realistic contact patch is necessary. For such a contact patch, the Air–Rubber ratio can be calculated along the leading and trailing edge. Through propagating these edges stepwise over the whole circumference of the tire, realistic changes in the Air–Rubber ratio are tracked. By Fourier transforming these values, spectra corresponding to psychoacoustic perceived tire noise can be calculated. Since the purpose of this research is not to describe two diverse tires exactly but to investigate the different methods of digitalization, a simplification is applied. The standard image is cropped symmetrically to the tire's centerline in each shoulder region. The remaining region is defined by minimal and maximal width, still included in Fig. 10.

The Air–Rubber ratio is calculated through

$$\text{Air–Rubber ratio} = \frac{\text{pixels}_{\text{white}}}{\text{pixels}_{\text{black}} + \text{pixels}_{\text{white}}} \quad (5)$$

Even though the total number of each pixel changes through different scaling of the pictures, the ratio remains the same, which makes the process robust.

### 5.2.2. Extraction of circumferential grooves

Circumferential grooves are especially important to avoid air being trapped in transverse grooves, which typically results in noisy tires. Additionally, they are also necessary for safety reason, as e.g. provision against aquaplaning. This makes them an important feature of each tire, including their width, location as well as depth. Following, a process to extract width as well as location of each circumferential groove is explained. The process is, as the shoulder tracking, based on the silhouette curve of the tire, presented in Fig. 10. The curve is once more smoothed and derived. From the central difference, points with low and high slopes are extracted. These basically define the beginning and end of a circumferential groove. Fig. 11 shows the smoothed outer shape, the related central difference and the corresponding image with pixel lines, highlighted in color, that are identified as locations with high and low slope. The grooves are extracted in the following manner: Going from high to low pixel values in  $x_2$ , the first two lines are selected. The absolute pixel value between these lines is calculated through adding up all pixel values in the enclosed area to one number. The resulting value is compared to the absolute pixel value obtained for the same area, assuming only white pixels. If the absolute pixel value is larger than 0.9 times the reference value, a groove is detected. In this case, the two lines are kept and the same process is done for the second and third line. If a groove is detected once more, the second line can be deleted, since line one and three form a wider groove. This process is repeated for every point detected through the central difference. The advantage of the combined approach between image and central difference over solely image processing lies in the customizability. Through the definition of a slope threshold that needs to be reached, grooves that are shallow will not be detected.

### 5.2.3. Profile randomization — pitch frequency

Looking at the transverse grooves, it is of importance whether the distance between grooves on each circumferential line are randomized as well as whether transverse grooves align for both shoulders or if there is a certain offset in-between grooves. The distance between grooves is referred to as pitch. While the randomization of the pitch does not necessarily reduce the overall emitted sound power, it spreads the power over a wider frequency band. This results in a less tonal and therefore more pleasant sound [46, p. 222–226]. Furthermore, this leads to a lower excitation at the resonance frequency which results in less sound emission. The offset in-between shoulders however results in lower force fluctuations in the contact area. This is equivalent to the effect for a change from straight to helical gears and results in lower broadband sound pressure levels.

To analyze this impact, the tire tread spectrum is calculated according to the method presented by Li et al. [11]. They used a laser system to create a flat 3D model of the tire with the dimensions width, length and profile height of the tread pattern. This is converted to time series by assuming a vehicle speed leading to the dimensions width, time and profile height. With this time series, a Fourier transformation is possible resulting in a frequency spectrum for each axial chunk of the tire. These spectra are merged into one coherent profile spectrum by summing up all complex Fourier coefficients over the width [11].

In the research presented in this publication, this process has to be adapted slightly. This is mainly the case because the principle of generating the 3D model is completely different. Li et al. [11] use a laser with defined resolution, leading to a structured grid pattern with constant spacing in width and length. The camera based approaches discussed in the presented research generates an unstructured tessellation mesh that does not guarantee constant spacing in width and length of the tire. This has to be taken into account before applying a Fast-Fourier-Transformation (FFT).

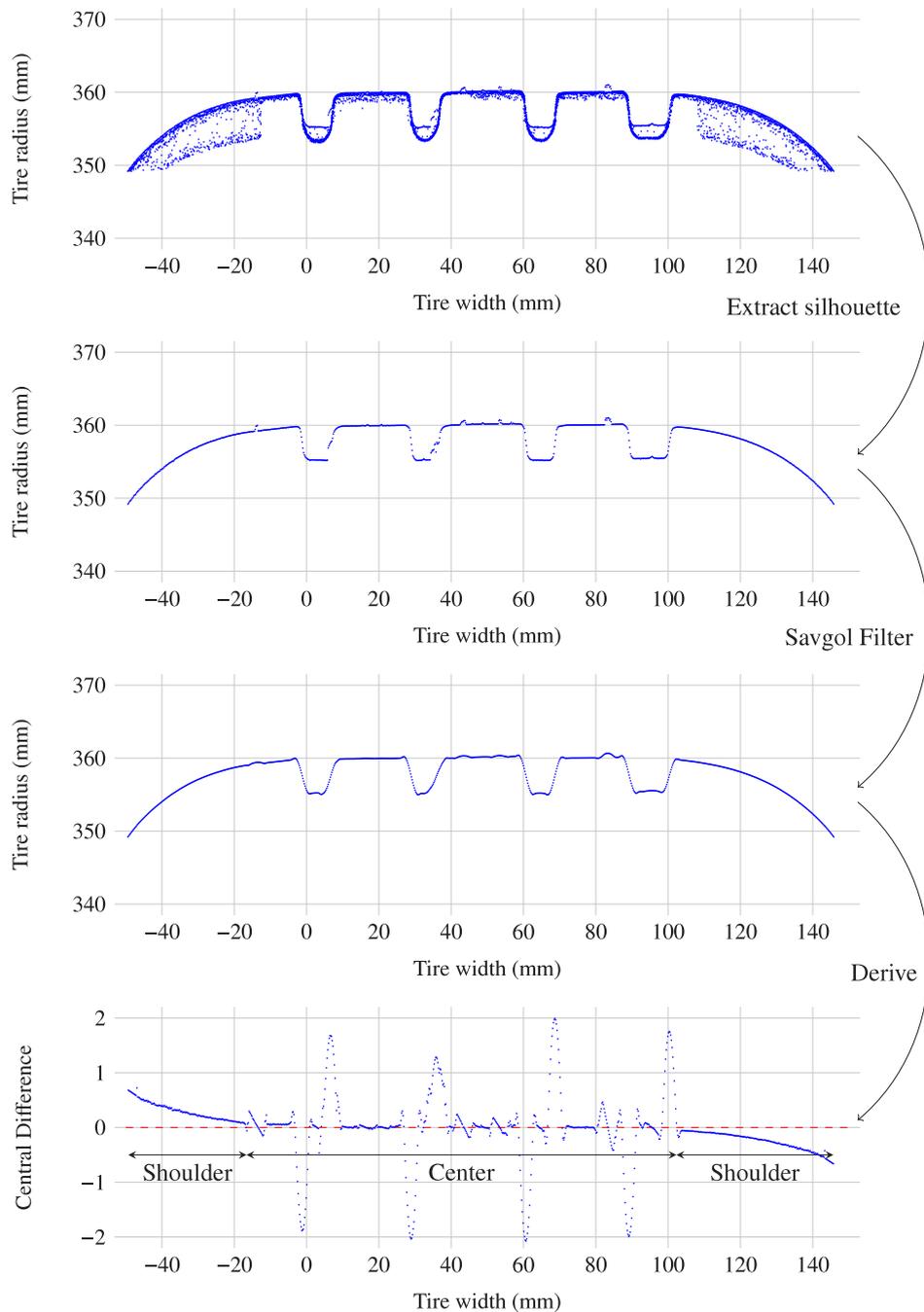


Fig. 10. Extracting the center region of the tire. For better visualization, the displayed points are downsampled by factor 150, 2, 2, and 2 for the graphs from top to bottom.

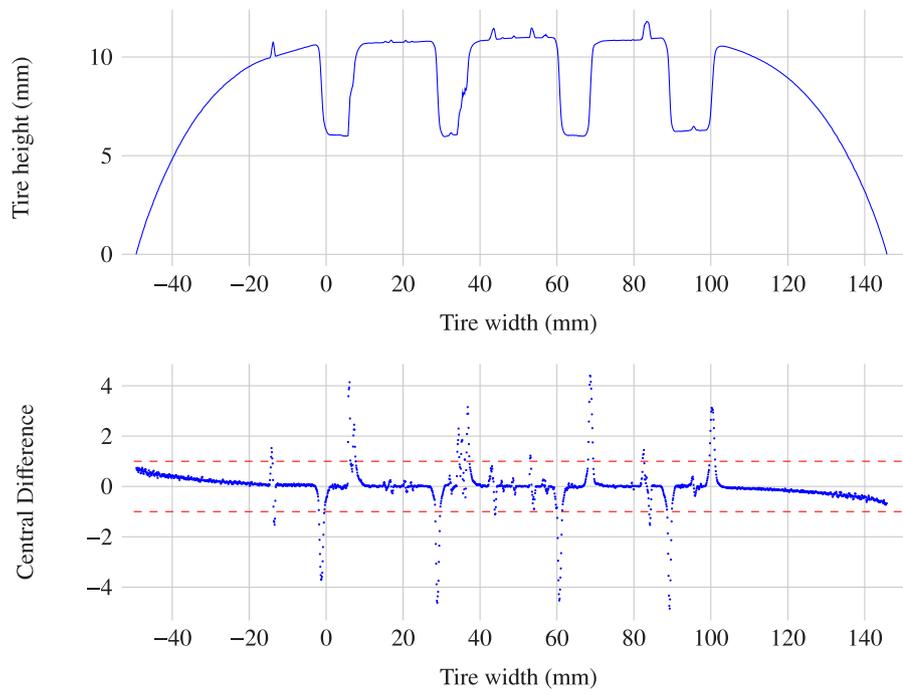
Due to that it is reasonable to resample the data points before applying the FFT. To enable a summation over the whole tire width afterwards, the resampling should be identical for each axial chunk. Therefore, the tire model is divided into circumferential chunks (as in Section 5.1). For all chunks, the number of grid points is counted. To avoid undersampling in the following step, the highest number of points in all chunks is selected as the baseline. Each circumferential chunk is resampled to twice the amount of these points via a linear interpolation. To make the spectra comparable, the summation is normalized with the sampling frequency before building the power spectral density.

Considering the tread pattern spectrum of one specific tire, a precise pitch frequency can be calculated. This is based on the assumption of a specific vehicle speed. This speed is then transformed into an

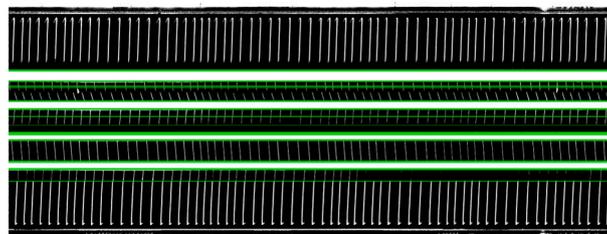
angular velocity and multiplied with the total number of transverse grooves on one shoulder. In this research, we investigate one summer and one winter tire, this leads to a pitch frequency of 417.8 Hz, assuming a speed of  $50 \text{ km h}^{-1}$  and 68 grooves. The spread bandwidth of the actual tire spectra received, around that pitch frequency describes how well a groove design is randomized.

## 6. Results

The following two subsections present the results obtained for the three tire features explained earlier on one summer and one winter tire, each compared for the two digitalization processes. For the open source pipeline, a normal digital single lens reflex camera was used.

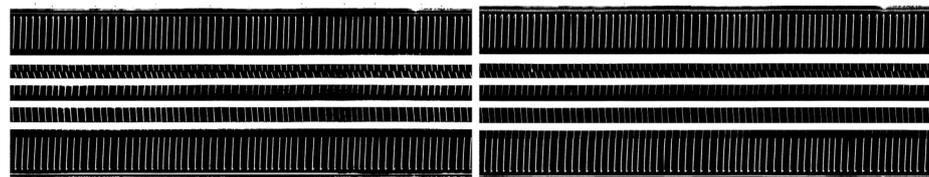


(a) Savgol filtered silhouette curve of a summer tire and corresponding central difference.



(b) Tread pattern image of a summer tire, green lines indicate positions where points exceed dashed red lines in central difference.

Fig. 11. Circumferential groove detection.



(a) Proposed model (b) Reference model

Fig. 12. Extracted standard tread pattern images.

6.1. Summer tire

As shown in Fig. 12, the summer tire leads in both digitalization processes to clear, visually comparable standard images.

The Air–Rubber ratio for equivalently cropped images results in comparable values with 0.282 for the reference model and 0.274 for the proposed model, which is a deviation of  $-2.8\%$ .

Out of both models, four circumferential grooves are extracted. The groove width as well as the center of the groove are displayed in Table 1. Since both digitalization processes do not guarantee the coordinate center to be at the identical location, relatively to the tire, the distance between grooves is of higher importance than their actual location. The reference and proposed models have deviations of 3.39%, 0.28%, and  $-0.53\%$  for the distances between neighboring grooves.

Table 1

Groove width and center location for the summer tire and different digitalization approaches.

Groove	Reference model width/center (mm)	Proposed model width/center (mm)	Deviation groove width (%)
1	7.86/3.93	7.82/3.20	$-0.5\%$
2	7.68/32.84	7.26/33.09	$-4.2\%$
3	7.73/64.68	7.77/65.02	$0.5\%$
4	10.98/94.91	11.07/95.09	$0.8\%$

Fig. 13 shows the tread pattern power spectra for both models, as well as the theoretical pitch frequency and its harmonics for the case of no randomization of the profile. These are labeled as 1, 2 and, 3. The large peak at low frequencies can be disregarded since it is mainly

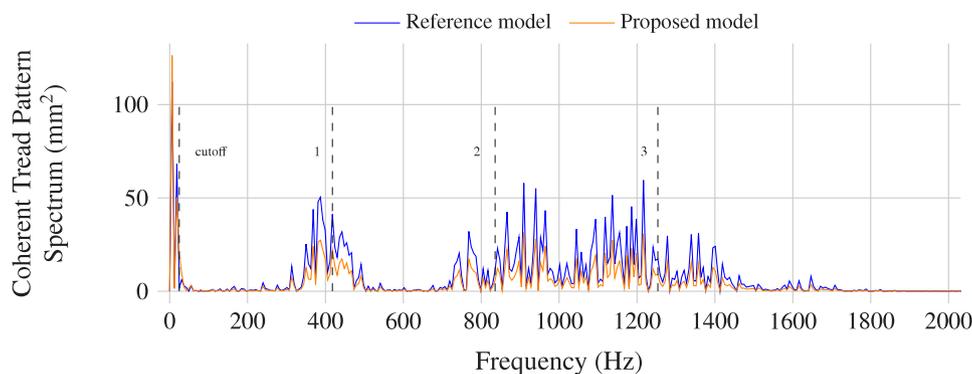


Fig. 13. Coherent tread pattern spectra for reference model (blue) and proposed model (orange) of the summer tire.

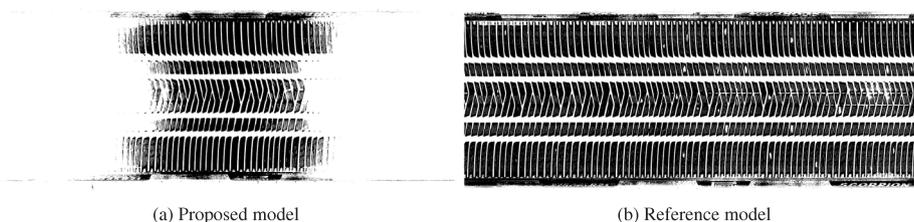


Fig. 14. Standard images of the winter tire for both digitalization processes.

a result of either measurement errors, leading to a slightly displaced center of the tire compared to the true physical center or a noncircularity of the tire which is the result of either manufacturing process, improper storage, or wear. The lower boundary frequency is defined as a signal appearing four times on the whole circumference, which lead to 24.6 Hz. Nevertheless, it is visible that in case of the summer tire, the FFT is able to detect the pitch frequency in both models. The overall spectra are highly comparable in their shape and only vary in absolute amplitude. This is not surprising, since the commercial system builds a tessellation surface model with higher resolution in the depth of the grooves and therefore also leads to higher peaks in the block profile.

### 6.2. Winter tire

The winter tire does not lead to comparable standard images for both modeling principles, as presented in Fig. 14.

The images are calculated with the same thresholds, nevertheless the proposed model does not lead to a full surface presentation of the tire. It rather shows a small section of the outer tire surface. Further variations in threshold as well as the chosen percentile of radius which defines the center of the threshold do not lead to more satisfying images. Decreasing the percentile mainly shifts the area of extracted surface. Meaning, decreasing the percentile leads to the grooves in the center of the image being closed while the border regions of the image are extracted. This behavior leads the question of whether the tire is perfectly round shaped. Looking at different cross section of the tire shows that the proposed tire model does not have a constant radius. As seen in Fig. 15, the difference between these cross sections ranges up to 4 mm. Considering the tire as a circle, the cross sections are taken at 0°, 90°, 180° and 270°. The difference in the radius is a result of a misplacement of the center point of the model.

The calculation of the Air–Rubber ratio for the proposed model is therefore done on only the part of the tire in which the tread pattern is fully represented. The obtained values are 0.370 and 0.351 for the reference and proposed model, which is a deviation of -5.1%. The groove widths of the winter tire with corresponding center locations are displayed in Table 2. The deviation between the models is slightly higher than in case of the summer tire. Looking at the center locations,

Table 2

Groove width and center point for the winter tire and different digitalization approaches.

Groove	Reference model width/center (mm)	Proposed model width/center (mm)	Deviation groove width (%)
1	7.46/-12.23	7.19/-64.87	-3.6%
2	8.28/18.69	7.70/-33.68	-6.9%
3	8.01/86.32	7.48/33.50	-6.6%
4	7.57/117.43	7.19/64.74	-5.0%

there is an almost constant offset between both models which results out of the different coordinate systems of the models. Nevertheless, the distance between the center of neighboring circumferential grooves remains comparable with 0.7%, -0.7%, and 0.4% deviation.

Regarding the average power spectrum of the winter tire in Fig. 16, leads to the same result as for the summer tire. The shapes and therefore randomization characteristics of the tread pattern profile, are detected similarly in both models. The large difference in amplitude for the winter tire results out of misplacement of the model's center. This assumption is supported through the previous observation, that the proposed winter model does not have a constant radius. Other reasons for the varying radius are implausible since these would also be detected in the reference model of the winter tire, which is not the case. The lower boundary frequency is defined as 22.4 Hz. Other than the similarity in the models' behavior, the winter tire shows a generally different output signal compared to the summer tire. While the signal analysis of the summer tire shows a clear first harmonic and a blurred second and third harmonic of the pitch frequency, the winter tire's shoulders are better randomized and offset from each other so that the harmonics almost cancel out. Nevertheless, it is to state that to calculate the shown power spectra for both winter and summer tire, the road is assumed to be smooth. In reality, the partly stochastic geometry of the road results in stochastic force fluctuation and therefore excitation without strong phase relationship on both shoulders. Due to changes in the acoustical impedance over the width of the tire, also known as horn effect, force fluctuations near the centerline of the tire have larger impact on overall sound pressure levels than fluctuations in the shoulder regions.

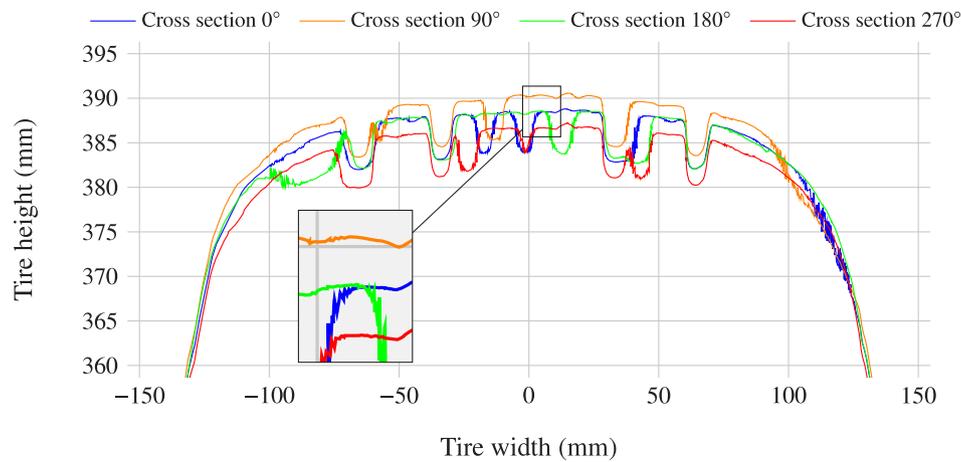


Fig. 15. Tire silhouette at four different cross sections of the winter tire from the proposed digitalization process.

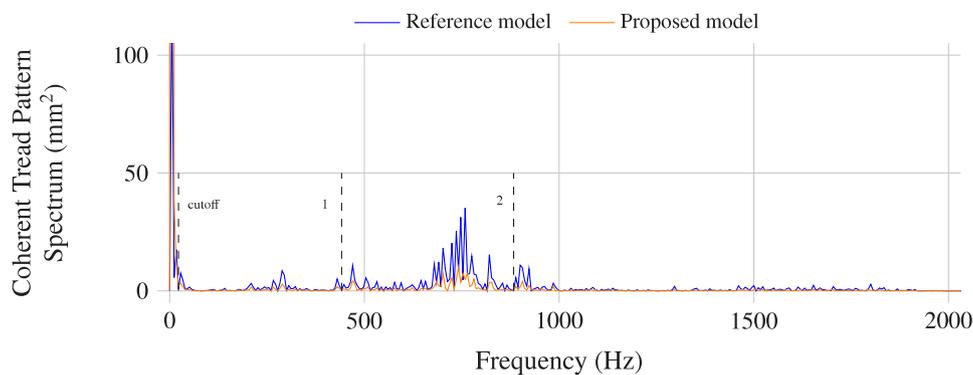


Fig. 16. Coherent tread pattern spectra for reference model (blue) and proposed model (orange) of the summer tire.

### 6.3. Calculation time

Other than the quality of the obtained models, the usability especially considering the necessary time to build a tire model can be compared.

Two steps can be investigated

1. taking images
2. calculation time

In the process of taking images, none of the systems has an advantage. To this point, the step wise rotation (in respect to  $x_2$  in Fig. 9) of the tire and changing of the angular orientation of  $x_2$  in respect to the camera system is done manually. While the commercial system needs fewer images than the proposed method to extract the shown results, taking one picture takes longer. This results from the applied version of the structured light approach, where different light patterns are projected onto the object. This essentially leads to a longer exposure time than the proposed method requires for each image. Nevertheless, these opposing trends cancel out for the shown models, resulting in an approximate time effort of 1.5 h per tire.

The necessary time for the reconstruction of the 3D models differs greatly between both systems. The commercial system runs on a 12 kernel (Intel Xeon Gold 6136) with 128 GB of RAM, where it takes around 30 min to reconstruct each tire model. The proposed method runs on a CPU with 6 kernels (Intel Core i7-1070H) with 64 GB of RAM, where it takes around 24 h for each tire [17].

The calculation time of the feature extraction depends only on the absolute number of grid points in the model. Since this is higher for

the commercial system, the necessary calculation time is also somewhat longer for this system. Nevertheless, the reconstruction of the standard images, as well as the calculation of the Air–Rubber ratio as well as the groove calculations happens within a few minutes. Only the tread pattern spectra calculation takes around 3 h. That time is needed since for each circumferential line, the grid points are first linearly interpolated to standardize the time step before applying the FFT. Since these lines are independent of each other, parallelization could speed up the calculation. The feature calculation is done on a 4 kernel CPU (Intel Core i7-6820HQ) with 48 GB of RAM.

## 7. Conclusion

In this paper we compare the tessellation surface model built from two different photogrammetry processes. One is constructed using a commercial application working with two different cameras as well as the structured light approach. The second one is constructed through a self built open source pipeline. The camera used for digitalization is a rather normal digital single-lens reflex camera with no special setting or ocular. The different approaches are used to digitize one summer and one winter tire each.

In the second part of the paper, three algorithms to extract characteristic features, describing tires and especially their acoustical behavior are explained. Even though there are many more relevant features, we decided to analyze the Air–Rubber ratio of the tire tread, the amount, width and location of circumferential grooves as well as a tread pattern frequency which also indicated the tread pattern's randomization.

These algorithms are applied on the four different tire models to compare the usability of the open source pipeline against the commercial system. In case of the summer tire, which generally have tread patterns with less lamella, the commercial system does not seem to have an advantage over the open source pipeline. The extracted standard footprint of the tire appears visually almost identical to the one of the commercial system. This is supported by the Air–Rubber ratio, which records a neglectable difference. Also, the circumferential grooves are recorded as almost same widths and locations. The tread pattern power spectra are also highly comparable, only varying in absolute amplitudes, which for the use case of this research is not of relevance. In case of the winter tire, the commercial system leads to an advantage. Especially the lamella are not really recorded in the open source model. This and an error in the absolute radius of the tire, lead to problems while extracting the standardized footprint. Nevertheless, comparing the Air–Rubber ratios in a smaller image frame, leads to similar values. Looking at the extracted tread pattern power spectra leads to similar results for both models. Both models show amplitudes in the same frequency ranges.

Future research should extract the tire features of a larger number of tires to link these statistically to the tire road noise. Since Ejsmont et al. [8] found six important tread pattern characteristics to be of great importance to the A-weighted sound pressure level, we assume that a dataset of 42 tire models will be enough to link the characteristics with supervised machine learning models to the sound pressure level. These measurements had already taken place and will be provided in further research. Also, the tire contact patch should be modeled in a more realistic shape than a rectangle. Considering the open source pipeline, more research is necessary to avoid the divergence in the radius and to extract lamella more precisely. Additionally, investigations into different modeling methods, especially using video material of tires, could be done. The presented feature extracting algorithms can also be applied to any other 3D representation of tires. Especially digital prototypes can be easily analyzed and their features can be used to predict e.g. the tires acoustic behavior.

### Abbreviations

The following abbreviations are used in this manuscript:

ANN	Artificial neural network
BPA	Ball-Pivoting Algorithm
MVS	Multi-View-Stereo
SfM	Structure-from-Motion
FFT	Fast-Fourier-Transformation

### CRedit authorship contribution statement

**Michael Leupolz:** Conceptualization, Methodology, Software, Validation, Formal analysis, Data curation, Visualization, Project administration, Writing – original draft, Writing – review & editing. **Timotheus Radler:** Methodology, Software, Validation, Data curation, Visualization, Writing – original draft. **Fank Gauterin:** Conceptualization, Supervision, Writing – review & editing.

### Data availability

The data that has been used is confidential

### Acknowledgments

The authors thank the Mercedes-Benz AG with the department overall vehicle integration NVH powertrain for their support. Additionally the authors thank Timo von Wysocki from the department overall vehicle integration NVH carbody and chassis for always being available to discuss the ongoing research. The authors thank Achim Winandi from the Karlsruhe Institute of Technology for valuable input and support.

### References

- [1] Evaluating 15 Years of Transport and Environmental Policy Integration: TERM 2015: Transport Indicators Tracking Progress Towards Environmental Targets in Europe, Vol. 2015, 7, EEA Report, Publications Office of the European Union, Luxembourg, 2015, URL: <http://publications.europa.eu/en/publication-detail/-/publication/6790e404-b509-11e5-8d3c-01aa75ed71a1>.
- [2] Noise and Health: Report By a Committee of the Health Council of the Netherlands, in: Publication / Gezondheidsraad, vol. 1994, 15E, Health Council of the Netherlands, The Hague, 1994.
- [3] UN/ECE Regulation No 51, Uniform provisions concerning the approval of motor vehicles having at least four wheels with regard to their sound emissions [2018/ 798], 2023, URL: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A42018X0798>. (Accessed 24 April 2023).
- [4] J. Putner, M. Lohrmann, H. Fastl, Contribution analysis of vehicle exterior noise with operational transfer path analysis, in: Proceedings of Meetings on Acoustics ICA2013, Vol. 19, 2013, 040035.
- [5] P. Zeller (Ed.), Handbuch Fahrzeugakustik: Grundlagen, Auslegung, Berechnung, Versuch, ATZ/MTZ-Fachbuch, third ed., Springer Vieweg, Wiesbaden and Heidelberg, 2018, <http://dx.doi.org/10.1007/978-3-658-18520-6>.
- [6] J. Jabben, E. Verheijen, C. Potma, Noise reduction by electric vehicles in the Netherlands, in: INTER-NOISE and NOISE-CON Congress and Conference Proceedings, 2012, pp. 6958–6965.
- [7] Springer Fachmedien Wiesbaden, “rollgeräusche kann man nicht mit emotionen verbinden”, ATZ - Automob. Z. 120 (2018) 18–21, <http://dx.doi.org/10.1007/s35148-018-0085-5>.
- [8] J.A. Ejsmont, U. Sandberg, S. Taryma, Influence of tread pattern on tire/road noise, SAE Trans. (1984) 632–640.
- [9] F. Stalter, F. Gauterin, Influence of circumferential tread pattern stiffness on tire road noise generation under driving torque, in: INTER-NOISE and NOISE-CON Congress and Conference Proceedings, Vol. 249, 2014, pp. 1149–1155.
- [10] T. Li, R. Burdisso, C. Sandu, The effects of tread patterns on tire pavement interaction noise, in: INTER-NOISE and NOISE-CON Congress and Conference Proceedings, Vol. 253, 2016, pp. 208–219.
- [11] T. Li, R. Burdisso, C. Sandu, An artificial neural network model to predict tread pattern-related tire noise, in: SAE Technical Paper Series, SAE International/400 Commonwealth Drive, Warrendale, PA, United States, 2017, <http://dx.doi.org/10.4271/2017-01-1904>.
- [12] R. Oorath, A. Saraswat, S.K. Gupta, N. Tiwari, S. Goyal, C. Patel, Influence of tread design parameters on air pumping noise in automotive tires, in: 24th ICSV 2017.
- [13] L. Rapino, L. Liu, A. Dinosio, F. Ripamonti, R. Corradi, S. Baro, Processing of tyre data for rolling noise prediction through a statistical modelling approach, Mech. Syst. Signal Process. 188 (2023) 110042, <http://dx.doi.org/10.1016/j.ymssp.2022.110042>.
- [14] C. Griwodz, S. Gasparini, L. Calvet, P. Gurdjos, F. Castan, B. Maujean, G. de Lillo, Y. Lanthony, Alicevision meshroom, in: Ö. Alay, C.-H. Hsu, A.C. Begen (Eds.), Proceedings of the 12th ACM Multimedia Systems Conference, ACM, New York, NY, USA, 2021, pp. 241–247, <http://dx.doi.org/10.1145/3458305.3478443>.
- [15] E. Rupnik, M. Daakir, M. Pierrot Deseilligny, Micmac – a free, open-source solution for photogrammetry, Open Geosp. Data Softw. Stand. 2 (2017) <http://dx.doi.org/10.1186/s40965-017-0027-2>.
- [16] E.-K. Stathopoulou, M. Welpner, F. Remondino, Open-source image-based 3d reconstruction pipelines: Review, comparison and evaluation, Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. XLII-2/W17 (2019) 331–338, <http://dx.doi.org/10.5194/isprs-archives-XLII-2-W17-331-2019>.
- [17] T. Radler, Development of a Tool for Generating a Surface Model from a Series of Images of Passenger Car Tires (Master’s thesis), Karlsruhe Institute of Technology, Karlsruhe, Germany, 2022.
- [18] M.J. Westoby, J. Brasington, N.F. Glasser, M.J. Hambrey, J.M. Reynolds, ‘Structure-from-motion’ photogrammetry: A low-cost, effective tool for geoscience applications, Geomorphology 179 (2012) 300–314, <http://dx.doi.org/10.1016/j.geomorph.2012.08.021>.
- [19] O. Özyeşil, V. Voroninski, R. Basri, A. Singer, A survey of structure from motion, Acta Numer. 26 (2017) 305–364, <http://dx.doi.org/10.1017/S096249291700006X>.
- [20] S. Bianco, G. Ciocca, D. Marelli, Evaluating the performance of structure from motion pipelines, J. Imaging 4 (2018) 98, <http://dx.doi.org/10.3390/jimaging4080098>.
- [21] X. Gao, T. Zhang, Introduction to Visual SLAM: From Theory to Practice, Springer Singapore, Singapore, 2021, <http://dx.doi.org/10.1007/978-981-16-4939-4>.
- [22] B. Kitt, Effiziente Schätzung dichter Bewegungsvektorfelder unter Berücksichtigung der Epipolarometrie zwischen unterschiedlichen Ansichten einer Szene, in: Zugl.: Karlsruhe, karlsruher institut für technologie (kit), diss., 2013, KIT Scientific Publishing, Karlsruhe, 2013, <http://dx.doi.org/10.5445/KSP/1000036444>.
- [23] Y.-J. Zhang, Handbook of Image Engineering, Springer Singapore, Singapore, 2021, <http://dx.doi.org/10.1007/978-981-15-5873-3>.
- [24] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, second ed., Cambridge University Press, Cambridge UK and New York, 2003.

- [25] J.L. Schönberger, Robust Methods for Accurate and Efficient 3D Modeling from Unstructured Imagery (Dissertation), Eidgenössische Technische Hochschule Zürich, Zürich, 2018, <http://dx.doi.org/10.3929/ethz-b-000295763>.
- [26] G. Van Rossum, F.L. Drake, Python 3 Reference Manual, CreateSpace, Scotts Valley, CA, 2009.
- [27] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, 2015, URL: <http://arxiv.org/pdf/1505.04597v1>.
- [28] P. Yakubovskiy, Segmentation models, 2019, URL: [https://github.com/qubvel/segmentation\\_models](https://github.com/qubvel/segmentation_models).
- [29] A. Dutta, A. Zisserman, The VIA annotation software for images, audio and video, in: Proceedings of the 27th ACM International Conference on Multimedia, MM '19, ACM, New York, NY, USA, 2019, <http://dx.doi.org/10.1145/3343031.3350535>.
- [30] A. Buslaev, V.I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, A.A. Kalinin, Albumentations: Fast and flexible image augmentations, Information 11 (2020) <http://dx.doi.org/10.3390/info11020125>, URL: <https://www.mdpi.com/2078-2489/11/2/125>.
- [31] C. Griwodz, S. Gasparini, L. Calvet, P. Gurdjos, F. Castan, B. Maujean, G.D. Lillo, Y. Lanthony, Alicevision meshroom: An open-source 3D reconstruction pipeline, in: Proceedings of the 12th ACM Multimedia Systems Conference, MMSys '21, ACM Press, 2021, <http://dx.doi.org/10.1145/3458305.3478443>.
- [32] J.L. Schö, J.-M. Frahm, Structure-from-motion revisited, in: Conference on Computer Vision and Pattern Recognition, CVPR, 2016.
- [33] J.L. Schönberger, E. Zheng, M. Pollefeys, J.-M. Frahm, Pixelwise view selection for unstructured multi-view stereo, in: European Conference on Computer Vision, ECCV, 2016.
- [34] P. Moulon, P. Monasse, R. Perrot, R. Marlet, OpenMVG: Open multiple view geometry, in: International Workshop on Reproducible Research in Pattern Recognition, Springer, 2016, pp. 60–74.
- [35] OpenMVS, 2022, URL: <https://github.com/cdcseacave/openMVS>. (Accessed 05 May 2022).
- [36] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, G. Taubin, The ball-pivoting algorithm for surface reconstruction, IEEE Trans. Vis. Comput. Graphics 5 (1999) 349–359, <http://dx.doi.org/10.1109/2945.817351>.
- [37] M. Kazhdan, M. Bolitho, H. Hoppe, Poisson surface reconstruction, in: Eurographics Symposium on Geometry Processing, 2006, p. 2006.
- [38] Q.-Y. Zhou, J. Park, V. Koltun, Open3D: A modern library for 3D data processing, 2018, arXiv:1801.09847.
- [39] T. Bell, B. Li, S. Zhang, Structured light techniques and applications, Wiley Encycl. Electr. Electron. Eng. (1999) 1–24.
- [40] P. Hehenberger, Computerunterstützte Produktion, Springer Berlin Heidelberg, Berlin, Heidelberg, 2020, <http://dx.doi.org/10.1007/978-3-662-60876-0>.
- [41] Carl Zeiss GOM Metrology GmbH, Triangulation: bewährtes messprinzip mit zukunft, 2023, URL: <https://www.gom.com/de-de/themen/triangulation>. (Accessed 06 January 2023).
- [42] APM Technologies 3D Pvt Ltd, Atos triple scan | industrial optical 3d digitizer, 2023, URL: [http://apmtech.in/atos\\_triple\\_scan.html](http://apmtech.in/atos_triple_scan.html). (Accessed 06 January 2023).
- [43] M. Schuth, W. Buerakov, Handbuch Optische Messtechnik: Praktische Anwendungen für Entwicklung, Versuch, Fertigung und Qualitätssicherung, in: Hanser eLibrary, Hanser, München, 2017, <http://dx.doi.org/10.3139/9783446436619>.
- [44] A. Savitzky, M.J.E. Golay, Smoothing and differentiation of data by simplified least squares procedures, Anal. Chem. 36 (1964) 1627–1639.
- [45] A. Sharma, Introduction To Computational Fluid Dynamics, Springer International Publishing, Cham, 2022, <http://dx.doi.org/10.1007/978-3-030-72884-7>.
- [46] U. Sandberg, J.A. Ejsmont, Tyre/Road Noise: Reference Book, first ed., INFORMEX Ejsmont & Sandberg Handelsbolag, Kisa, Sweden, 2002.