Acta Cybernetica **26** (2024) 581–591.

Extracting Line Parameters of Woven Wire Mesh in Images under Directional Illumination*

László Körmöczi[ab] and László G. Nyúl[ac]

Abstract

Localizing the wires of a mesh in an image is important in various image processing applications. This task can be difficult if the wires cannot be detected with simple line detectors, e.g. if corrugated wires of a woven mesh appear as dark and bright segments under directional illumination. Template matching is insufficient if the appearance of the wires varies throughout the image, depending on the viewing angle, and neural networks require computationally expensive training on a well-prepared dataset. We propose an efficient way to extract the line parameters (position and orientation) of the wires of a regular mesh from an image by finding meaningful local minima of a cost function, followed by RANSAC-controlled robust outlier filtering.

Keywords: image processing, wire detection, cage mesh detection, line parameter extraction

1 Introduction

Detecting wire mesh in an image and extracting the line parameters (i.e. the location and orientation of the wires' projection in the image) is an important task in various image processing applications. Localising the wires can help in 2D-3D camera pose estimation, or for inpainting. This is useful when the relative pose of the camera and the wire mesh (e.g. an animal cage) can change and has to be known throughout a series of images or a video stream. In such applications, if a wire mesh (e.g. the front mesh of the cage) is visible in the image, it can be used for reliable, unsupervised camera pose estimation if a subject (e.g. rodent) is to be localized in a coordinate system in which the mesh is fixed.

A mesh made of straight wires can easily be detected with line detectors, e.g. using Canny edge detector [1] followed by Hough transform [2]. Detection of thick

*Project no. TKP2021-NVA-09 has been implemented with the support provided by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the TKP2021-NVA funding scheme.

[a]Department of Image Processing and Computer Graphics, University of Szeged, Hungary
[b]E-mail: kormoczi@inf.u-szeged.hu, ORCID: 0000-0002-3833-0609
[c]E-mail: nyul@inf.u-szeged.hu, ORCID: 0000-0002-3826-543X

DOI: 10.14232/actacyb.299046

lines can be achieved by either downscaling the image or by using thick line detectors [8, 3]. However, these methods fail when the wires are not seen as lines in the image, like the wires of a welded corrugated mesh that appear as wavy lines with alternating darker and brighter segments due to directional reflection of their material.

Different parts of a mesh of corrugated wires are seen from different angles, as in the examples in Fig. 1, thus have different shape throughout the image, so template matching cannot be used.

Figure 1: Examples of corrugated wires seen from different angles.

Quasi-periodic wire meshes can be detected in the frequency domain [5]. There are methods that use machine learning and neural networks to detect and segment meshes [10], but training neural networks requires a lot of carefully prepared training data (e.g. precisely segmented samples) and is computationally expensive.

Wire mesh detection can be easily achieved if there is camera motion between frames [9] or the focusing distance of the lens can be varied [12], but having a fixed-focus camera in a fixed position requires a different approach.

If the wire mesh fills a large part of the image, we can find "bands" that can be macroscopically recognised by an average intensity, although having a large local variation. The regularity of the mesh (i.e. the distance between wires is constant and the mesh is a rectangular grid) can be utilized without the need to rectify the image. In this work, we extend the procedure described in [7] and show quantitative results on test images.

Detection and localisation of a wire mesh is needed in several image processing applications. In medical experiments, rodents are often used as models for human diseases. For behavior analysis, the animals are placed in a cage and observed with cameras outside the cage [6]. Localisation in the image is possible but the subject's location is of interest in the cage coordinate system. Reliable pose estimation of the camera with respect to the cage is needed and can be achieved using the front wire mesh of the cage.

2 Line parameter extraction for wires

In order to find the line parameters of the wires' projection in the image, we calculate a cost function in a 2-dimensional parameter space, then find strong local minima (negative spikes) of the cost function as candidates for detected lines, and finally apply a robust filtering on the candidates.

2.1 Preprocessing

The algorithm can find projections of wires in a grayscale image that fit in a rectangle ("band") of arbitrary position and orientation. If the source image suffers from lens distortion (e.g. barrel or pincushion distortion), an undistortion step has to be performed. Perspective effect does not have to be compensated, since straight lines remain straight after perspective projection.

The image should be normalized so that the pixel values are in the $[0, 1]$ range.

2.2 Cost function calculation

First, an average intensity value $\overline{I}$ is to be defined that macroscopically represents the wires in the image, along with a band width w that covers a wire. We compute the absolute difference of pixel intensity and $\overline{I}$ for each pixel of the image I as:

$$J(x, y) = |I(x, y) - \overline{I}|$$

Then the image space of J is transformed into a (ρ, θ) parameter space with a cost function, where ρ denotes the distance of a line from the image center and θ denotes the rotation of the line. The calculation is performed separately for finding the projection of vertical and horizontal wires, and θ represents the deviation from the vertical or horizontal direction, respectively.

The cost function C is calculated for (ρ, θ) pairs, ρ ranging from $\rho_{\min}$ to $\rho_{\max}$ with step size ρ_{step} and θ ranging from $\theta_{\min}$ to $\theta_{\max}$ with step size θ_{step}. The $C(\rho, \theta)$ value of C at given ρ and θ is calculated as the sum of the intensity of the pixels of J that are covered by a w wide band around a line that is at ρ distance from the image center, rotated with θ (denoted as $B_{w,\rho,\theta}$), divided by the area of the image covered by the band ($|B_{w,\rho,\theta}|$):

$$C(\rho, \theta) = \frac{\sum\limits_{(x,y) \in B_{w,\rho,\theta}} J(x, y)}{|B_{w,\rho,\theta}|}$$

In this approach, the possible values of the cost function lie within the $[0, 1]$ range, 0 represents a band that has $\overline{I}$ intensity in each pixel. Furthermore, we do not consider bands that cover less than half of the area of a vertical band (for the vertical case) or a horizontal band (for the horizontal case). The function value of such points of the parameter space are set to 1.

2.3 Finding candidates for lines

Good candidates are negative peaks of C. A negative peak can be defined as being a local minimum in its (large enough) neighbourhood and deeper than a threshold compared to its neighbouring baseline.

As C is calculated in a given finite range with a finite step size for both parameters, this ordered set of calculated values can be treated as an image, with axes ρ and θ, and the pixel intensities are values of C at given ρ and θ values.

We declare the neighbourhood in which the local minimum search is performed as a rectangle $\mathcal{N}_{d\rho,d\theta}$ with axes along the parameter space dimensions. For $\mathcal{N}_{d\rho,d\theta}(\rho,\theta)$, the maximal distance between the (ρ,θ) point and any point in $\mathcal{N}_{d\rho,d\theta}(\rho,\theta)$ is $d\rho$ and $d\theta$ along the two dimensions:

$$\mathcal{N}_{d\rho,d\theta}(\rho,\theta) = \{\, (\rho',\theta') \mid |\rho'-\rho| \leq d\rho \wedge |\theta'-\theta| \leq d\theta \,\}$$

To fulfill the aforementioned two criteria (i.e. having local minimum and sufficiently small value compared to a baseline), we choose (ρ,θ) pairs by two conditions. We select those in a set $\mathcal{L}$ that have local minima in $\mathcal{N}_{d\rho,d\theta}$ neighbourhood:

$$\mathcal{L} = \{\, (\rho,\theta) \mid C(\rho,\theta) = \min_{(\rho',\theta')\in\mathcal{N}_{d\rho,d\theta}(\rho,\theta)} C(\rho',\theta') \,\}$$

We independently perform a bottom-hat (black-hat or black top-hat) transform on C treated as an image (as described above) with a rectangle as structuring element defined by $\mathcal{N}_{d\rho,d\theta}$, followed by thresholding. Bottom-hat transform performs closing on an image, then subtracts the original image from the closed image, thus extracts the baseline and transforms negative peaks into positive [11]. For simplicity, denote the structuring element by $\mathcal{N}$ and the image representation of C values by C. Let C' be the result of the bottom-hat transform:

$$C' = (C \bullet \mathcal{N}) - C = ((C \oplus \mathcal{N}) \ominus \mathcal{N}) - C$$

(where $\bullet$ denotes closing, $\oplus$ is dilation and $\ominus$ is erosion).

Thresholding (with a c_t threshold) applied to the result of the bottom-hat transform selects (ρ,θ) pairs in a set $\mathcal{V}$ that are part of a valley:

$$\mathcal{V} = \{\, (\rho,\theta) \mid C'(\rho,\theta) > c_t \,\}$$

Points of the parameter space that fulfill both criteria are selected as candidates in a set $\mathcal{P}$:

$$\mathcal{P} = \mathcal{L} \cap \mathcal{V}$$

2.4 Robust filtering

As there is only a perspective projection present in the image after correcting for optical distortion, and a regular mesh consists of two (usually perpendicular) sets of parallel wires that are to be detected independently, the lines for the image of the parallel wires are either parallel or intersect at a vanishing point $V(u,v)$. For lines that intersect at $V(u,v)$, the following is true for every line, if ρ denotes the distance of a line from the image center and θ denotes the rotation of the line:

$$\rho = u\cos\theta + v\sin\theta$$

If a vanishing point exists for a set of lines, solving a linear system for two (ρ,θ) value pairs referring to two of these lines gives an explicit result, except in the case

when $|\theta_1 - \theta_2| = 180°$ (where θ_1 and θ_2 belong to the first and second selected line, respectively) and the equation system has infinite solutions. In this configuration, the two lines coincide and the vanishing point can be any point on that line.

If the lines in the image are parallel, there is no solution for the equation system, as there is no vanishing point and the (ρ, θ) pairs representing these lines lie on a line in the parameter space.

Although the above function is highly non-linear, for many practical applications, when the vanishing point is outside the image, the (ρ, θ) points representing the visible lines in the image fit on a line with a small tolerance. Line fitting also works for those configurations when the equation system does not have one exact solution.

We use RANSAC [4] to fit a line on the candidates. RANSAC is widely used in numerous applications for fast and robust selection of inliers, because it can reliably filter out outliers and is robust for low inlier ratio. We assume that outliers do not fit another line (e.g. no other strongly visible grid-like structure is present in the image).

Algorithm 1 Line parameter extraction for wires

Input: $\overline{I}$ average intensity of the wires in the image
w band width
I grayscale image
H, W height and width of the image
$\theta_{\min}$, $\theta_{\max}$, θ_{step} as minimal, maximal rotation angle and rotation step size
$\rho_{\min}, \rho_{\max}, \rho_{\text{step}}$ as minimal and maximal signed distance from image center and distance step size
Output: ρ, θ line parameters of the wires in the image
for $x \in [0, W]$, $y \in [0, H]$ **do**
$J(x, y) = |I(x, y) - \overline{I}|$
end for
for $\theta \in [\theta_{\min}, \theta_{\max}]$ with θ_{step} step size **do**
for $\rho \in [\rho_{\min}, \rho_{\max}]$ with ρ_{step} step size **do**
Let $B_{w,\rho,\theta}$ be a w pixel wide band that is at ρ distance from the image center and rotated with θ
$C(\rho, \theta) = \frac{\sum_{(x,y) \in B_{w,\rho,\theta}} J(x,y)}{|B_{w,\rho,\theta}|}$
end for
end for
Search for candidates with $\mathcal{N}_{d\rho,d\theta}$ neighbourhood:
$\mathcal{L} = \{ (\rho, \theta) \mid C(\rho, \theta) = \min_{(\rho', \theta') \in \mathcal{N}_{d\rho,d\theta}(\rho,\theta)} C(\rho', \theta') \}$
$\mathcal{V} = \{ (\rho, \theta) \mid C'(\rho, \theta) > c_t \}$, where $C' = (C \bullet \mathcal{N}) - C = ((C \oplus \mathcal{N}) \ominus \mathcal{N}) - C$
$\mathcal{P} = \mathcal{L} \cap \mathcal{V}$
Line fitting and outlier filtering with RANSAC

3 Experimental results

In the experiments described in [6] we are working with images of 60×60×60 cm rodent home cages acquired with consumer grade IP cameras having 1/2.7" class sensors and wide angle lens with 3.6 mm focal length. The resolution of the images is 1920×1080 px and the cage almost fills the whole image. Each cage is observed by two cameras in a non-standard vertical stereo configuration, facing the front of the cage. The angle between the cameras' optical axis and the axis perpendicular to the front wire mesh is around 20-30°. The cages are placed on a movable platform so that the relative pose of the cameras and the cage can change. The cameras are mounted on rotatable heads, thus the stereo configuration can also change. The cameras have 83° vertical and 44° horizontal angle of view and a strong barrel distortion can be observed in the images. We calibrated our cameras and undistorted the images so that barrel distortion is eliminated. Fig. 2 shows that only a perspective effect remains.

Figure 2: Sample original (left) and undistorted (right) image showing the rodent home cage

The home cages have a skeleton of square steel tubes and a mesh of corrugated wires is welded to the inner side of the tubes.

We ran the proposed algorithm on 25 images, 13 taken in daylight conditions with artificial directional illumination and 12 at night, with the cameras set to night vision mode and inbuilt infrared LEDs illuminated the scene. The expected width w was set to 9 px and we observed the effect of varying $\overline{I}$ in the range from 0 to 1.0 with 0.05 step size. The resolution of the parameter space was 1 px for ρ and 0.2° for θ, ρ swept through the image in both dimensions and θ was limited between -15° and 15°. $d\rho$ was set to 40 px and $d\theta$ was set to 8° for $\mathcal{N}_{d\rho,d\theta}$ used in Section 2.3. Tolerance in RANSAC filtering for a candidate to fit on a line was set to 2° and 10 px.

Filtered candidates (predicted positives) were compared against ground truth values that were computed from manually assigned lines for each wire in the image. A predicted positive is considered true positive if θ error is no more than 1° and ρ error is no more than 5 px.

Precision, recall and F_1 score (average ± standard deviation) are shown in Tables 1 and 2, for vertical and horizontal lines, respectively. Although the test images

originate from 4 cameras, we aggregated the results because the conditions were similar for each camera. With $\overline{I} < 0.6$ there were no correctly detected wires in most images, so we excluded these results from the table.

We can see from the results that the algorithm is robust for a large variation of $\overline{I}$, and it can detect almost all wires of the mesh if $\overline{I}$ is set correctly. The total number of vertical lines to be detected was 24, and the number of horizontal lines to be detected was 12 or 14 in most cases. Results for the horizontal wires are better, as some of the vertical wires near the side of the cage were not detected correctly due to the background and the side meshes. False positives generally come from the skeleton of the cages.

Table 1: Precision, recall and F_1 score for finding vertical lines.

	daylight			night		
	precision	recall	F_1 score	precision	recall	F_1 score
$\overline{I}$	(avg $\pm$ std)	(avg $\pm$ std)	(avg $\pm$ std)	(avg $\pm$ std)	(avg $\pm$ std)	(avg $\pm$ std)
0.60	0.32 ± 0.03	0.48 ± 0.06	0.38 ± 0.04	0.25 ± 0.05	0.42 ± 0.07	0.31 ± 0.06
0.65	0.51 ± 0.04	0.66 ± 0.06	0.58 ± 0.04	0.35 ± 0.06	0.48 ± 0.08	0.41 ± 0.07
0.70	0.67 ± 0.06	0.76 ± 0.03	0.71 ± 0.03	0.48 ± 0.06	0.56 ± 0.05	0.51 ± 0.06
0.75	0.76 ± 0.05	0.81 ± 0.04	0.78 ± 0.04	0.61 ± 0.05	0.69 ± 0.05	0.65 ± 0.05
0.80	0.80 ± 0.04	0.82 ± 0.03	0.81 ± 0.03	0.65 ± 0.09	0.72 ± 0.06	0.69 ± 0.08
0.85	0.85 ± 0.04	0.84 ± 0.01	0.84 ± 0.03	0.68 ± 0.06	0.76 ± 0.05	0.72 ± 0.06
0.90	0.83 ± 0.03	0.82 ± 0.01	0.83 ± 0.01	0.69 ± 0.06	0.77 ± 0.05	0.73 ± 0.06
0.95	0.82 ± 0.04	0.82 ± 0.01	0.82 ± 0.02	0.70 ± 0.07	0.78 ± 0.05	0.74 ± 0.06
1.00	0.82 ± 0.05	0.81 ± 0.02	0.81 ± 0.03	0.70 ± 0.06	0.78 ± 0.04	0.74 ± 0.05

Table 2: Precision, recall and F_1 score for finding horizontal lines.

	daylight			night		
	precision	recall	F_1 score	precision	recall	F_1 score
$\overline{I}$	(avg $\pm$ std)	(avg $\pm$ std)	(avg $\pm$ std)	(avg $\pm$ std)	(avg $\pm$ std)	(avg $\pm$ std)
0.60	0.46 ± 0.06	0.70 ± 0.03	0.55 ± 0.04	0.38 ± 0.05	0.83 ± 0.12	0.52 ± 0.07
0.65	0.53 ± 0.03	0.71 ± 0.01	0.61 ± 0.03	0.47 ± 0.03	0.87 ± 0.09	0.61 ± 0.04
0.70	0.62 ± 0.08	0.81 ± 0.06	0.71 ± 0.07	0.62 ± 0.07	0.89 ± 0.09	0.73 ± 0.06
0.75	0.77 ± 0.09	0.89 ± 0.04	0.83 ± 0.07	0.84 ± 0.09	0.94 ± 0.05	0.88 ± 0.05
0.80	0.84 ± 0.12	0.90 ± 0.04	0.87 ± 0.08	0.92 ± 0.11	0.92 ± 0.06	0.91 ± 0.06
0.85	0.88 ± 0.07	0.90 ± 0.03	0.89 ± 0.03	0.94 ± 0.07	0.92 ± 0.06	0.93 ± 0.03
0.90	0.90 ± 0.03	0.90 ± 0.03	0.90 ± 0.02	0.96 ± 0.07	0.91 ± 0.06	0.93 ± 0.01
0.95	0.93 ± 0.00	0.90 ± 0.03	0.92 ± 0.02	0.97 ± 0.04	0.90 ± 0.06	0.94 ± 0.02
1.00	0.93 ± 0.00	0.90 ± 0.03	0.92 ± 0.02	0.97 ± 0.04	0.89 ± 0.05	0.93 ± 0.01

An example visualization can be seen in Figs. 3 and 4. Fig. 3 shows the values of the cost function over the selected (ρ, θ) range for horizontal lines of a test image, with $\overline{I} = 0.95$. Brighter color means higher value, 0 is black and 1 is white. Negative peaks can be observed as dark spots. After filtering the candidates, inliers that fit on a line are marked with green, while outliers are marked with blue. Fig. 4 shows found lines painted over an example image, horizontal lines are red, vertical lines are green.

We use the proposed method as part of a video processing pipeline, and intersection points of the lines are used for 2D-3D relative pose estimation (i.e. the transformation between the camera coordinate system and the cage coordinate system). For that application, not all wires are required to be detected, but the accuracy of the line parameters is crucial.

Figure 3: Example visualization of C cost function for horizontal wires. Horizontal axis is ρ and vertical axis is θ. Brighter color means higher function value. Top: Negative peaks can be seen as dark spots. Bottom: All candidates as described in Section 2.3, filtered as in Section 2.4. Inliers are marked with green, outliers are marked with blue.

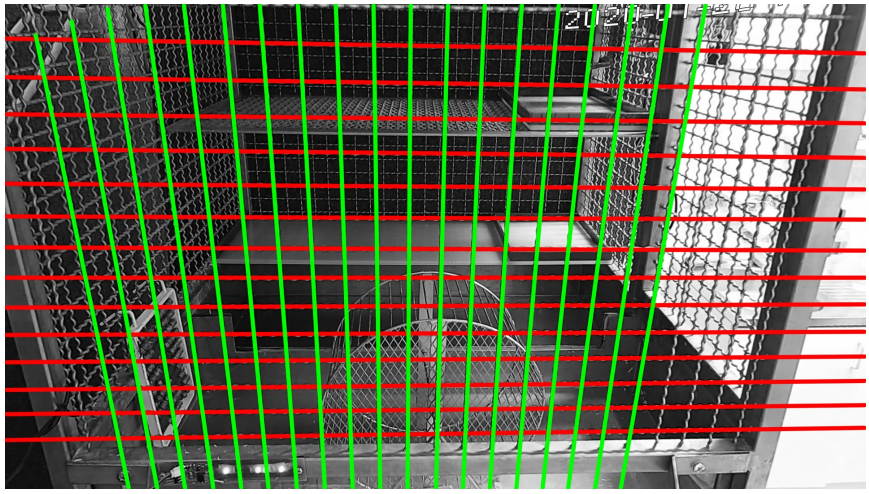

Figure 4: Found lines painted over the original image, vertical lines in green and horizontal lines in red.

The parameter w has to be determined for a given experimental setup. The proposed method is not sensitive to this parameter and can tolerate deviations that occur in most cases. A problem can arise when there is a large angle between the image plane and the plane of the mesh, resulting in significantly different distances between the camera and the two edges of the mesh. As a result, the thickness of the wires in the image varies within a wide range. In such situations, a band with a given width can cover more than one wire in some parts of the image while covering only a small part of a single wire in other areas.

As described in Section 1, other methods cannot be used reliably and easily on our image set. Line detectors cannot be applied, as the wires do not appear as lines in the image. For Hough transform, precise binarization of the image would be needed with a carefully set threshold. However, on our images, especially in night conditions, a strict threshold results in almost all wires except at the center of the image not being detected at all. Conversely, by using a more permissive threshold, the wires of the side and back of the cage, along with other objects, make detection of the front wires impossible. An example visualization for a night image is presented in Fig. 5. Pre-trained models of neural network-based methods are trained on wire mesh samples that differ from the ones present in our images, and cannot detect the mesh. Training would require a huge number of manually segmented samples.

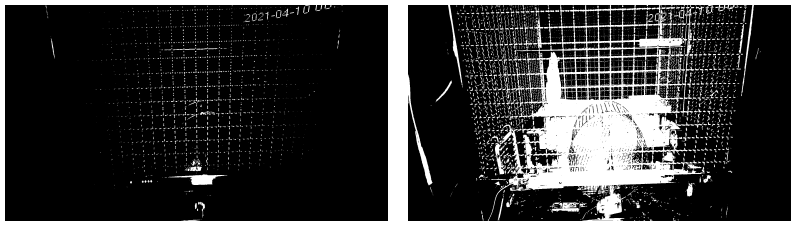

Figure 5: Illustration of problems with binarizing for Hough transform shown on a night image. Left: wires except the central ones disappear with a restrictive threshold. Right: wires of the side and back of the cage, along with other objects, make the detection of the front wires impossible with a more permissive threshold.

4 Conclusion

We presented an efficient method for extracting line parameters of the projection of wires of a woven mesh in an image by transforming the image space into a 2D parameter space and finding and robustly filtering local minima of the resulting cost function. Experimental results show that the algorithm is able to accurately detect wires and filter out false detections in general experimental setups where the vanishing point is outside the image.

Although the presented method works robustly for the desired application, several improvement possibilities could be investigated. An optimizer could be utilized

to find local minima so that the computational cost could be reduced if the cost function is not calculated for the entire parameter space. We plan to examine the effect of running the method on gradient images instead of the original images, so that $\overline{I}$ would not have to be defined. We also plan to incorporate a solver for the linear equation system described in Section 2.4 to make the algorithm usable for meshes that are seen from a low angle.

References

[1] Canny, J. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986. DOI: 10.1109/tpami.1986.4767851.

[2] Duda, R. and Hart, P. Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972. DOI: 10.1145/361237.361242.

[3] Even, P., Ngo, P., and Kerautret, B. Thick line segment detection with fast directional tracking. In *Image Analysis and Processing*, pages 159–170. Springer International Publishing, 2019. DOI: 10.1007/978-3-030-30645-8_15.

[4] Fischler, M. and Bolles, R. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. DOI: 10.1145/358669.358692.

[5] Hettiarachchi, R., Peters, J., and Bruce, N. Fence-like quasi-periodic texture detection in images. *Theory and Applications of Mathematics and Computer Science*, 4(2):123–139, 2014. URL: http://cs.umanitoba.ca/~bruce/papers/periodic.pdf.

[6] Körmöczi, L., Kalmár, G., Adlan, L., Büki, A., Kékesi, G., Horváth, G., and Nyúl, L. Rágcsálók viselkedésmintázatának kutatása automatizált videóelemzéssel. In *Képfeldolgozók és Alakfelismerők Társaságának 13. konferenciája*, 2021. 17 pages, URL: https://kepaf.njszt.hu/kepaf2021/submissions/submission_00047.pdf.

[7] Körmöczi, L. and Nyúl, L. Detecting corrugated wire mesh in images. In *Proceedings of the 13th Conference of PhD Students in Computer Science*, pages 116–120. University of Szeged, 2022. URL: https://www.inf.u-szeged.hu/~cscs/cscs2022/pdf/cscs2022.pdf.

[8] Lo, R.-C. and Tsai, W.-H. Gray-scale Hough transform for thick line detection in gray-scale images. *Pattern Recognition*, 28(5):647–661, 1995. DOI: 10.1016/0031-3203(94)00127-8.

[9] Lueangwattana, C., Mori, S., and Saito, H. Removing fences from sweep motion videos using global 3d reconstruction and fence-aware light field rendering. *Computational Visual Media*, 5(1):21–32, 2019. DOI: 10.1007/s41095-018-0126-8.

[10] Matsui, T. and Ikehara, M. Single-image fence removal using deep convolutional neural network. *IEEE Access*, 8:38846–38854, 2020. DOI: 10.1109/access.2019.2960087.

[11] Serra, J. *Image analysis and mathematical morphology*. Academic Press, 1982. ISBN: 0126372403.

[12] Yamashita, A., Matsui, A., and Kaneko, T. Fence removal from multi-focus images. In *Proceedings of the 20th International Conference on Pattern Recognition*. IEEE, 2010. DOI: 10.1109/icpr.2010.1101.