BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 69(3), 2021, Article number: e137121
DOI: 10.24425/bpasts.2021.137121

ARTIFICIAL AND COMPUTATIONAL INTELLIGENCE

Multi-line signal change detection for image segmentation with application in the ceramic tile industry

Filip SUŠAC*, Tomislav MATIĆ, Ivan ALEKSI, and Tomislav KESER

J. J. Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek,
Kneza Trpimira 2B, 31000 Osijek, Croatia

Abstract. In the ceramic industry, quality control is performed using visual inspection in three different product stages: green, biscuit, and the final ceramic tile. To develop a real-time computer visual inspection system, the necessary step is successful tile segmentation from its background. In this paper, a new statistical multi-line signal change detection (MLSCD) segmentation method based on signal change detection (SCD) method is presented. Through experimental results on seven different ceramic tile image sets, MLSCD performance is analyzed and compared with the SCD method. Finally, recommended parameters are proposed for optimal performance of the MLSCD method.

Key words: segmentation; edge detection; biscuit tile; image processing; visual inspection; ceramic industry.

1. Introduction

Nowadays, modern ceramic tile companies are applying the most sophisticated technologies of robotics and computer information to improve the step of automation. Therefore, almost all production stages are completely automated, except for the stage of visual inspection [1]. During ceramic tile production, three products are made: green tile (body), biscuit tile (body with decoration), and finally ceramic tile (finished product). Visual inspection on the production line can be carried out in all three stages, and therefore reduce the number of defective final products. Due to labor costs, most companies conduct only visual inspection of the final product (ceramic tiles), which is inefficient in terms of material recycling, energy consumption, green economy, and financial efficiency [2].

Currently, quality control of the final product can be divided into three phases. In phase one, the tile arc is measured by a linear planer. In the second phase, tile size is measured by stacker and compared to ideal tile size. In the last phase, ceramic tile surface defects are identified by human workers. Usually, the quality of tiles is graded by a human operator according to ISO 10545-2 standard. However, problems like sickness, lack of attention, eye fatigue, and lover repeatability often result in misclassification caused by human factors [3].

Automatic quality control of the biscuit/ceramic tiles consists of several stages: image acquisition, image preprocessing and segmentation, defect detection, and finally classification. To develop a completely automated system for visual inspection, a deep analysis of the complex ceramic tile production process needs to be conducted to determine which defects can be found in each stage of visual inspection. Nevertheless, in every stage tile image segmentation is necessary as it affects the performance of the subsequent methods for quality control. As aforementioned, the complete production of the ceramic tile is divided into three stages. Firstly, the green tile is formed using a material powder mixture and a large press machine. Subsequently, the biscuit tile is formed by adding glazing and/or decoration on the green tile. The biscuit tile can be easily damaged during fabrication as it is not fired yet [4]. The last production stage is the ceramic tile made by firing a biscuit tile in the kiln. In this process, surface defects as depigmentation, spots, cracks, blobs, edges, and pinholes are most common [1,5]. To better understand the complexity of ceramic tile production, a complete production process is given in Fig. 1 [3].

The rest of this paper is organized as follows. Section 2 gives the literature overview of the paper topic. Section 3 is divided in subsections where, signal change detection (SCD) method is described in 3.1, new multi-line signal change detecion (MLSCD) method is presented in 3.2 and proposed segmentation method is given in 3.3. Experimental analysis of the proposed segmentation method is given in Section 4. Finally, Section 5 concludes the paper and provides remarks for future work.

2. Related work

In the past decade, numerous papers are published in the field of automatic visual inspection, as almost every branch of the production industry demands some kind of visual inspection to classify the final product. The complete process of visual inspection can be divided into three parts. First is the digitalization of tested objects using some digital device such as cameras or lasers. The second is image analysis, where image segmentation is a crucial part. Finally, the third step is the quality classi-

*e-mail: filip.susac@ferit.hr

Manuscript submitted 2020-08-10, revised 2021-03-08, initially accepted for publication 2021-04-07, published in June 2021

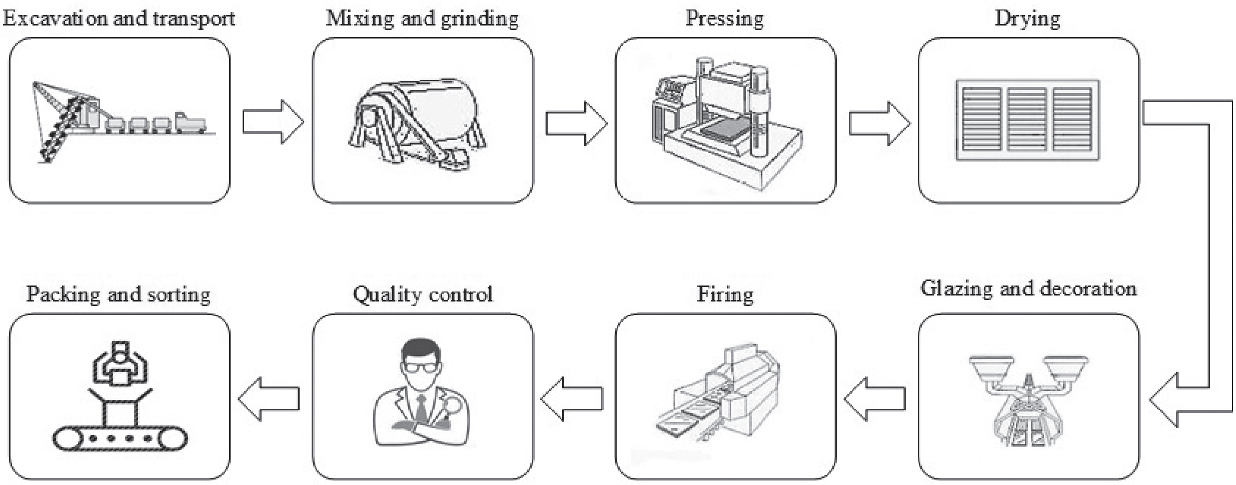


Fig. 1. Ceramic tile production process

fication [6]. Nowadays, the focus is placed on the machine and deep learning, whereas in the past it was on statistical and analytical methods. A combination of the both are still commonly used in numerous real-world applications [7].

In the field of visual inspections and the classification of ceramic tiles, authors use different methods. Wang et al. in [8] use Mask region-based convolutional neural network for segmentation and damage detection in historic glazed tiles with the aim to develop an automatic periodic control system. Which plays a crucial role in the maintenance and protection of the existing infrastructure. In [9] authors use the K-means clustering method to detect the temperature of the firing zone to improve flexural strength and fracture toughness of ceramic tiles. Matic et al. in the paper [10] present real-time edge and corner defects detection systems. As the proposed system is based only on white uniform tiles, segmentation is done by threshold using Otsu's method. A simple and efficient method for surface defects is presented by Hocenski et al. in the paper [11]; the proposed method is also applied in uniform white ceramic tiles, and for tile segmentation, authors use Sobel edge detector. The proposed method is reused in paper [12] with three different implementations: CPU, GPU, and FPGA to measure and compare execution times for different computational platforms. During computer vision inspection, the working environment can drastically impact the performance of the algorithm. So Zguang et al. in their paper [13] present a robust segmentation algorithm based on super-pixel. The proposed algorithm is tested on several different industry vision inspection input images, one of which is the ceramic industry. Rahaman and Hossain in paper [1] present automatic defect detection and classification system, similar to [11] they use Sobel edge detection method. Zhou et al. proposed a new method for image segmentation based on the least-square circle detection and entropy rate super-pixel. Although the method is applied in the glass bottle industry, a similar method has potential uses in the ceramic tile industry [14]. Yan et al. in paper [15] present another similar industry application of image preprocessing. In their paper, fast and effective framework, applied on the hot round steel surface is given, where they use parallel realization of the Sobel edge detector for effective region extraction.

Quality inspection of ceramic tiles is conducted using another approach, such as Chuna et al. use in their paper [16]. They proposed a non-destructive testing method based on two different machine learning methods where input data is not an image but rather its ceramic tile sound produced by the piston.

Matic et al. [17] proposed real-time biscuit tile image segmentation (BTS) using SCD method. In their paper, the proposed BTS method is extensively compared with eight different biscuit tile segmentation methods (AEM-GC, OTSU, ACWE, MCET, RF, HMRFEM, MET, GF), and the BTS method outperformed them all. This information is used to further improve the segmentation process and compare the results.

An improvement of the SCD-based BTS method is the main motivation for writing this manuscript. In this work, a new statistical method for image segmentation of biscuit tiles based on the SCD method is presented. Improving the existing SCD method from single to multi-line calculation we create a new multi-line SCD (MLSCD) method. The MLSCD method is implemented and tested on seven biscuit tiles datasets and evaluated using three different evaluation methods.

3. Proposed method

3.1. Signal Change Detection (SCD) method. In signal processing, statistical methods are commonly used to detect amplitude jumps. One of them is the SCD method. Authors in [17] use the SCD method to find a single jump in a 1D signal of pixels. The jump occurs from the background to the tile image pixels.

Let $\mathbf{x}$ be the 1D stochastic signal, and x_i is a single sample of $\mathbf{x}$, where $i = 1, \ldots, N$ is sample index, and N is the number of samples. The SCD method return pairs of indices n_j and amplitudes A_j that approximates $\mathbf{x}$. The proposed method can be used to find unknown amplitude jump $\Delta A = A_{j+1} - A_j$ at an unknown index n_j [18].

Let $\tilde{\mathbf{x}}$ be the approximated signal of $\mathbf{x}$, to detect a single jump from background pixel level to the tile pixel level $\Delta A = A_t - A_b$. The jump occurs in $\mathbf{x}$ at an unknown jump index n_e. In this way, an approximation $\tilde{\mathbf{x}}$ of 1D signal $\mathbf{x}$ is done with

$$\tilde{x}_i = \begin{cases} A_b, & i = 1, \ldots, n, \\ A_t, & i = n+1, \ldots, N. \end{cases} \tag{1}$$

In Eq. (1), the jump index n represents the result of proposed segmentation method in a single signal. Two amplitude levels, the background pixels average value A_b, and the tile pixels average value A_t, are calculated with

$$A_b = \frac{1}{n}\sum_{i=1}^{n} x_i, \qquad A_t = \frac{1}{N-n}\sum_{i=n+1}^{N} x_i\,. \tag{2}$$

In this work, the least square method is used to find the optimal jump index n that minimizes the sum of two squared residuals

$$V_{SCD}(n) = \sum_{i=1}^{n} (x_i - A_b)^2 + \sum_{i=n+1}^{N} (x_i - A_t)^2\,. \tag{3}$$

In this way, the problem is represented as finding the minimum variance before and after the jump index n. Finally, the *optimal jump index* n_e that minimizes the sum of two variances, cf. Eq. (3) is defined with

$$n_e = \underset{1 \le n \le N-1}{\arg\min}\, V_{SCD}(n). \tag{4}$$

3.2. MLSCD method. Equations (3) and (2) describe the problem of finding the jump index n_e, cf. Eq. (4), in signal x_i, i.e. a single row of pixels. This paper generalizes the problem of finding the jump index n_e in 1D signal by considering multiple adjacent image rows at once as a signal $x_{i,j}$. The method is rotationally invariant and the same approach can be applied to image columns. Therefore, Eqs. (4), (3) and (2) are generalized with

$$n_K = \underset{\substack{K \in \{2k-1:\, k \in \mathbb{W}\} \\ 1 \le n \le N-1}}{\arg\min}\, V_{SCD}(n, K), \tag{5}$$

$$\begin{aligned} V_{SCD}(n,K) = {} & \sum_{j=-\lfloor K/2 \rfloor}^{\lfloor K/2 \rfloor} \sum_{i=1}^{n} (x_{i,j} - A_b(K))^2 + \\ & + \sum_{j=-\lfloor K/2 \rfloor}^{\lfloor K/2 \rfloor} \sum_{i=n+1}^{N} (x_{i,j} - A_t(K))^2, \end{aligned} \tag{6}$$

and

$$\begin{aligned} A_b(K) &= \frac{1}{nK} \sum_{j=-\lfloor K/2 \rfloor}^{\lfloor K/2 \rfloor} \sum_{i=1}^{n} x_{i,j}\,, \\ A_t(K) &= \frac{1}{(N-n)K} \sum_{j=-\lfloor K/2 \rfloor}^{\lfloor K/2 \rfloor} \sum_{i=n+1}^{N} x_{i,j}\,, \end{aligned} \tag{7}$$

where $K \in \{2k-1:\, k \in \mathbb{W}\}$ is the odd number of adjacent neighbour rows that are considered at once. Row index j is ranging from -$\lfloor K/2 \rfloor$ to $\lfloor K/2 \rfloor$ with values relative to 0. Relative index number $j = 0$ corresponds to the current row for which the Eqs. (6) and (7) are calculated. Subsequently, the first and the last $\lfloor K/2 \rfloor$ image rows are discarded from the equation since there are no adjacent neighbour rows available at the start and at the end of an image. Brackets $\lfloor . \rfloor$ denote the floor rounding operation.

Finally, to segment the background from the foreground, the MLSCD approximates the signal $x_{i,j}$ with $\tilde{x}_i$, $i = 1, \ldots, N$, for each row of the image. Indices $j = -\lfloor K/2 \rfloor, \ldots, \lfloor K/2 \rfloor$ are considered to calculate the variance in the signal $x_{i,j}$ with its neighbour image rows. The signal $\tilde{x}_i$ has two jump levels $A_b(K)$ and $A_t(K)$, cf. Eq. (7). Jump levels $A_b(K)$ and $A_t(K)$ are defined by the K number of adjacent neighbour rows, and by the mean value of $x_{i,j}$, before and after the sample n, respectively.

3.3. Biscuit tile image segmentation using MLSCD. In this paper, MLSCD is used for offline segmentation purposes. In the first step, an input image is converted from RGB to the grayscale color space. MLSCD parameter N that represents the total number of background pixels p_b and tile pixels p_t is determined using Matic et al. [17]. The result of the SCD and MLSCD methods depends on the number of total pixels N, and on the necessary number of tile pixels p_t. For a given N, p_t must fulfill the condition $0.2N \le p_t \le 0.8N$ for the best segmentation result. In this paper, $N = 400$ fulfills the given condition for all images in the dataset.

In the next step, grayscale input image is divided in four parts: I_{left}, I_{top}, I_{right} and I_{bottom}

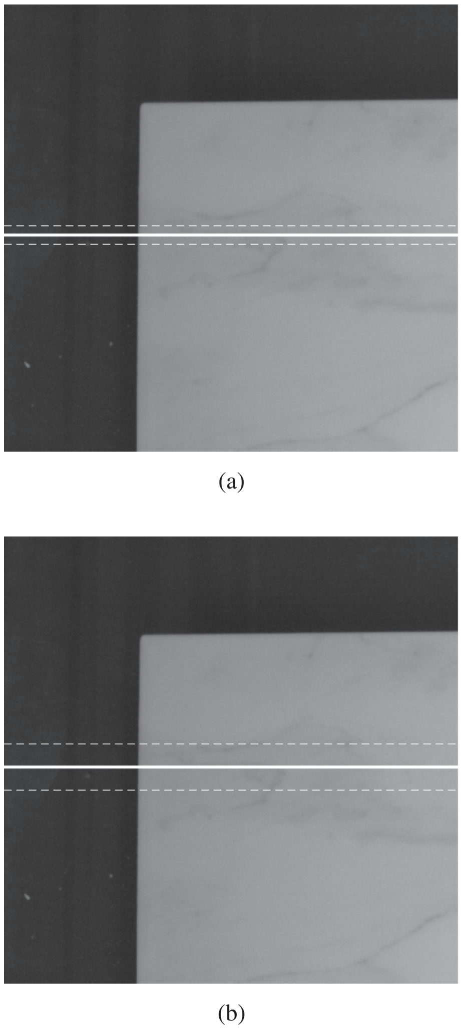
$$I_{left} = \begin{bmatrix} p_x(1,1) & \cdots & p_x(N,1) \\ \vdots & \ddots & \vdots \\ p_x(1,H) & \cdots & p_x(N,H) \end{bmatrix}, \tag{8}$$

$$I_{right} = \begin{bmatrix} p_x(W-N,1) & \cdots & p_x(W,1) \\ \vdots & \ddots & \vdots \\ p_x(W-N,H) & \cdots & p_x(W,H) \end{bmatrix}, \tag{9}$$

$$I_{top} = \begin{bmatrix} p_x(1,1) & \cdots & p_x(W,1) \\ \vdots & \ddots & \vdots \\ p_x(1,N) & \cdots & p_x(W,N) \end{bmatrix}, \tag{10}$$

$$I_{bottom} = \begin{bmatrix} p_x(1,H-N) & \cdots & p_x(W,H-N) \\ \vdots & \ddots & \vdots \\ p_x(1,H) & \cdots & p_x(W,H) \end{bmatrix}, \tag{11}$$

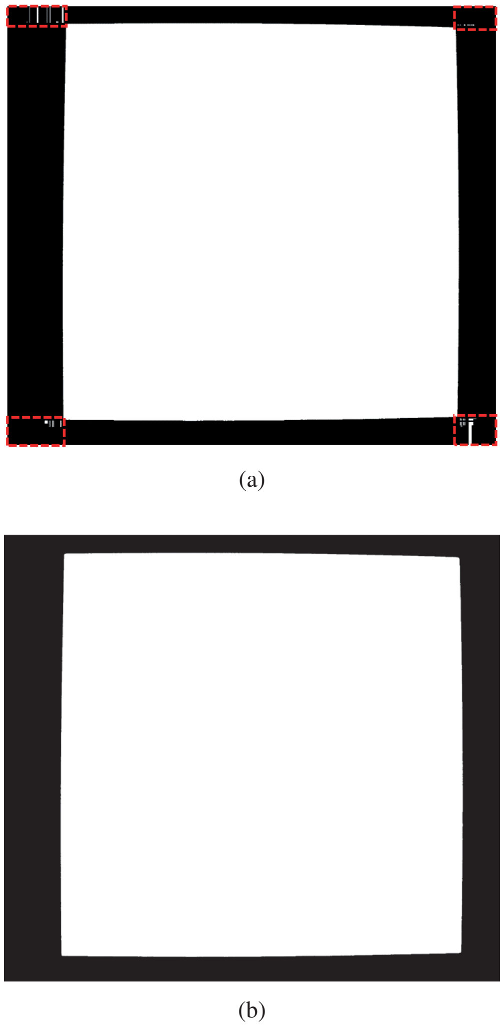
where $p_x(column, row)$ represents the pixel location with the origin in the top left corner of an image. Each image part is processed independently with the selected N and K values. The intermediate result is the list of n_K indices for each image part: $n_{K,left}$, $n_{K,right}$, $n_{K,up}$ and $n_{K,down}$, where n_K represents the found jump index for each row/column, cf. Fig. 2.



(a)

(b)

Fig. 2. Example of used rows for MLSCD calculation of the left part of the image: $K = 9$ (a); $K = 21$ (b)



(a)

(b)

Fig. 3. Binary image with small white region (outliers) in the image corners marked with red dashed rectangles (a); the final MLSCD segmentation result (b)

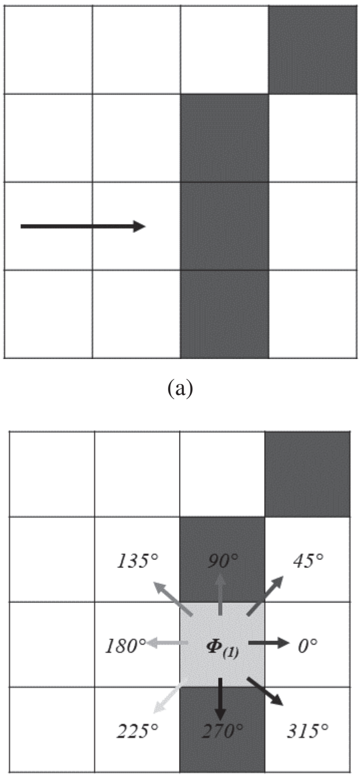
After combining the list of indices n_K for each image part, a binary image is created with the following procedure. In each row/column, a total of two jump indices is detected. The background pixels are defined as $B = B_{L,R} \bigcup B_{U,D}$, where the background pixels for image rows are $B_{L,R} = \left[0, \ldots, n_{K,left}\right] \bigcup \left[n_{K,right}, \ldots, W\right]$, and the background pixels for image columns are $B_{U,D} = \left[0, \ldots, n_{K,up}\right] \bigcup \left[n_{K,down}, \ldots, H\right]$. The output segmented binary image is formed in a way that the background pixels B are set to the value 0 (black color), and the rest of the image pixels correspond to tile pixels $T = |B - 1|$ with the pixel value 1 (white color). The result of the described segmentation procedure is a black and white image. However, after combining the results from both rows and columns, a single large white region is produced, which represents tile pixels exclusively. Outliers are present in a form of small, pixel size patches of white regions in the image corners, cf. Fig. 3a red dashed rectangles mark the outliers. To discard those unwanted corner regions of the binary image, the largest connected component is calculated for the created binary image using the union-find algorithm [19]. Finally, the result of the proposed method is a binary image that consists of background (black) and tile (white) pixels, cf. Fig. 3b.

3.4. Extraction of contour and contour descriptor

3.4.1. Contour extraction. After the MLSCD segments the tile pixels from background pixels, the segmented binary image I_S is provided. To get some additional information about the segmentation performance, it is necessary to find the contour around the tile edge. This information can also be used subsequently in edge and corner defect detection [20]. To find the tile contour, firstly, a binary erosion operation is applied to the image I_S. The result is a binary image with a smaller tile pixel area. The image I_S is then subtracted from the eroded I_S to get the binary contour image I_C, i.e.

$$I_C = I_S - \left(I_S \ominus \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \right). \qquad (12)$$

As a result of Eq. (12), the contour image I_C contains a pixel-wide edge of the segmented tile image I_S [21]. The image I_C is used to extract the contour in its 2D vector form. To determine the start position of the contour, the contour finding method used in this work starts at the position in the first (leftmost) column, and at the center row, i.e. at $p_x(1, \lceil H/2 \rceil)$, cf. Fig. 4a. The first position of the contour is found by searching for the first detected tile pixel edge in I_C, starting from the $p_x(1, \lceil H/2 \rceil)$ to the right side of the image I_C. The first detected edge pixel $p_x(c_e, \lceil H/2 \rceil)$ is the start position of the contour, cf. Fig. 4b. From the first contour position, a 90° direction is used to find the next contour location. The next location is found by searching the 8-neighborhood in the clockwise direction, starting from a relative shift of $-135°$ from the previous direction, cf. Fig. 4b. This procedure is continued until the starting point is reached. Finally, the contour positions are obtained as a vector of 2D positions.


(a)

(b)

Fig. 4. Contour descriptor finding method starts from the $p_x(1, \lceil H/2 \rceil)$ to the right direction (a); first detected edge with initial direction angle set to 90°, and its 8-neighbourhood clockwise searching directions from 225° (light-gray arrow) to 270° (black arrow) (b)

3.4.2. Contour descriptor extraction. The contour descriptor is obtained from the contour positions by calculating the set Φ of absolute direction (angle) differences between the two adjacent contour positions. The contour descriptor has 4 characteristic angle differences of 90°, where each jump represents one tile corner. Between two tile corners, along a tile edge, there are many spikes of 45° angle difference, cf. Fig. 5. The spikes are caused by the following three reasons: tile alignment, edge defects, and mostly by the image segmentation performance. The first reason is caused by an alignment between a camera and a tile, cf. Fig. 9a. The second reason indicates the presence of an edge defect. Finally, the third reason, with the largest share in the total number of spikes, is the image segmentation performance. Better performance results with a smaller number of spikes, cf. Fig. 9.

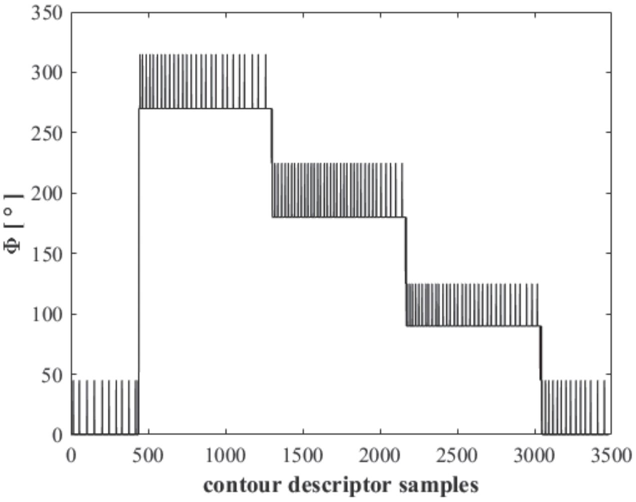


Fig. 5. Contour descriptor Φ of Bianco Mat ground truth image

4. Experimental results

4.1. Data acquisition and preparation. All biscuit tile images are acquired with a computer vision station (CVS) shown in Fig. 6 [22]. The CVS is a device designed and produced for a production line in the ceramic tile factory Keramika Modus d.o.o. It is placed on the production conveyor line after the decoration stage and before the firing stage. Ink-Jet printing

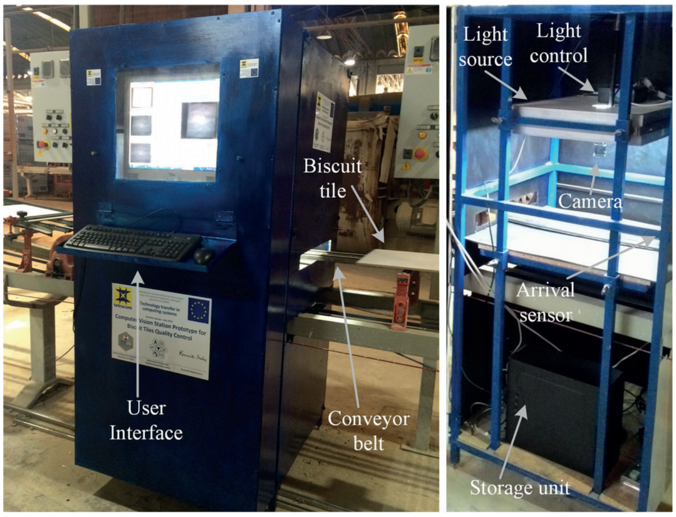


Fig. 6. Computer vision station prototype for biscuit tile image acquisition

technology is used for a raw tile decoration [23]. It is a non-contact decoration method that facilitates the production of high-quality prints using a wide range of materials. Main components of the CVS are shown in Fig. 6. It consists of a metal housing with air filtering fans, an LED light system, an area-scan color camera, a storage (processing) unit, a tile detection sensor, and a screen with a user interface. The used camera has a maximum resolution of 2592 × 2048 pixels and a maximum frame rate of 60 fps. The camera is connected to the processing unit with the USB 3.0 protocol. All captured color images are saved on the processing unit hard drives in compressed *.jpg* format. Biscuit tiles are not affected in any way by the computer vision station prototype. After a tile image is acquired, it continues its way to the firing kiln.

CVS is used for the formation of the image dataset used for the experimental analysis. The dataset consists of 7 biscuit tile image sets, where each set contains 6 biscuit tile images, making the dataset with a total of 42 images. All dataset images represent biscuit tiles with no defects. Image examples of biscuit tiles are depicted in Fig. 7. Tiles Botticinio, Vellagio, Sabuni Beige, and Madielle Dark, cf. Fig. 7, consist of six similar but different patterns. Tile design is not uniform for the aforementioned tiles.

Using GIMP 2.10.8 image editing software [24], a human expert created a ground truth binary image for each image in the dataset.

4.2. Segmentation evaluation. Segmented image with a smaller number of misclassified pixels is perceived as connatural to the ground truth image. The SCD image segmentation method (MLSCD with $K = 1$) usually results with misclassified pixels at positions in the vicinity of tile edges, cf. Fig. 9.

To evaluate these misclassified pixels and overall segmentation results, three different measures for qualitative and quan-

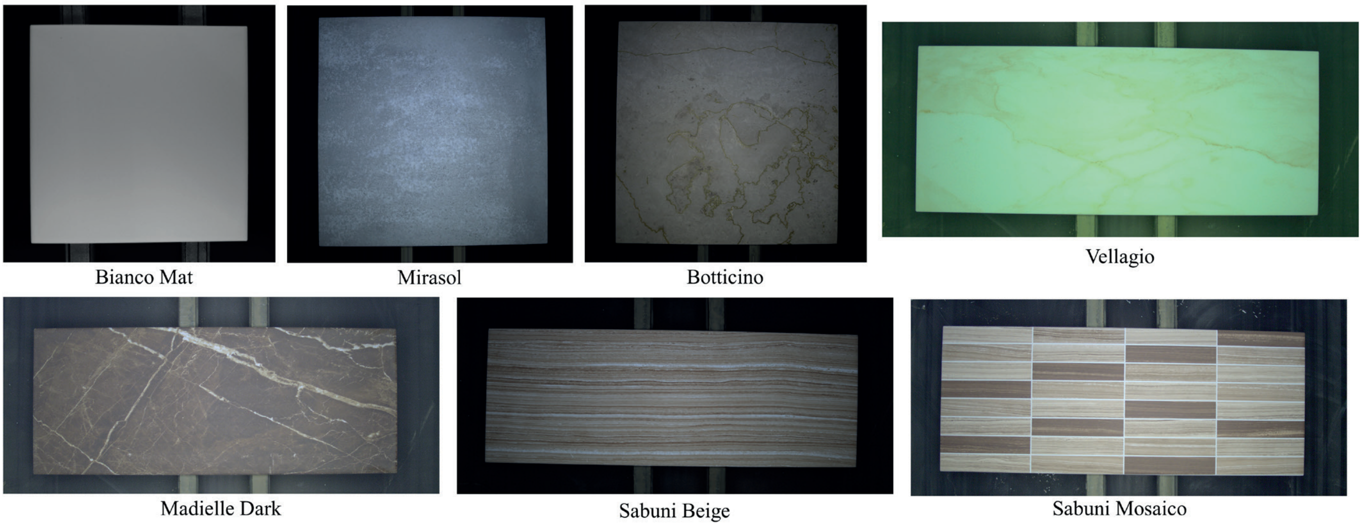


Fig. 7. One tile example from each set used for experimental analysis

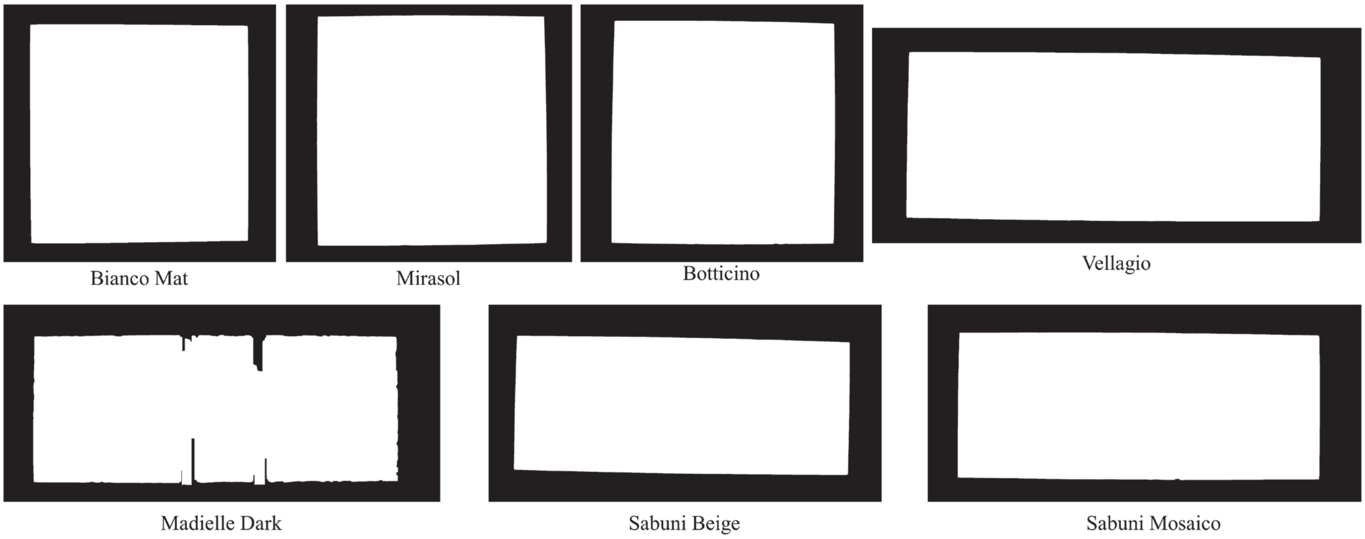


Fig. 8. MLSCD results for $K = 25$ of the tile examples from Fig. 7

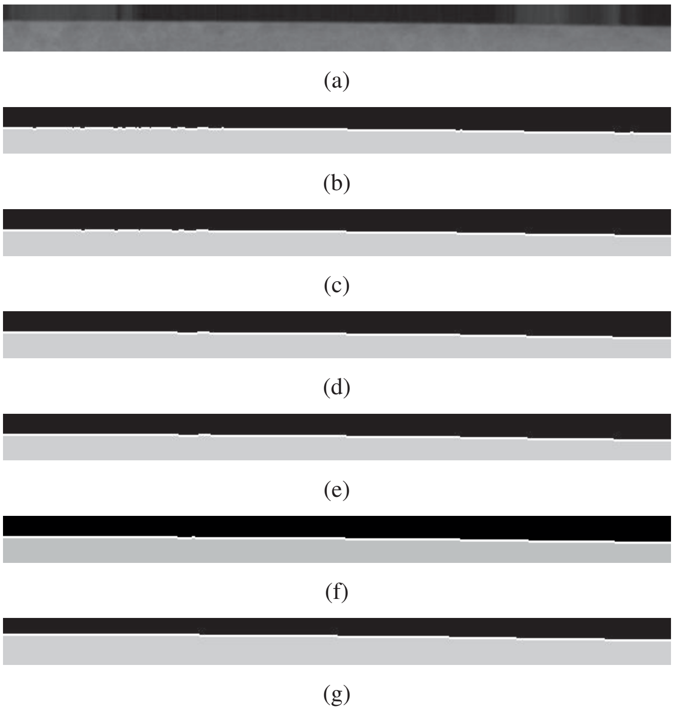


Fig. 9. Top edge part of one biscuit tile image from Mirasol image set: input grayscale image (a); MLSCD segmentation image with $K = \{1, 7, 15, 19, 25\}$ (b–f); ground truth image (g)

titative evaluation are introduced in this paper: a new measure number of small deviations (NSD), the pixel-based misclassification (PBM), and a new measure contour angles comparison (CAC).

The first qualitative evaluation measure is a new NSD measure. It is defined as the number of small pixel deviations (corresponds to spikes on the Fig. 5) near the edge of the tile pixels. It is calculated using the Algorithm 1 based on the contour descriptor Φ, cf. Section 3.4.2.

Algorithm 1 Number of Small pixel Deviations (NSD)

```
procedure NSD($\Phi$)
    $count \leftarrow 0$
    for $i = 2$ to $|\Phi|$ do
        $\Delta\phi \leftarrow |\Phi(i) - \Phi(i-1)|$
        if ($\Delta\phi = 45$ or $\Delta\phi = 315$) then
            $count \leftarrow count + 1$
        end if
    end for
end procedure
```

For results evaluation, two NSD values are calculated: Normalized NSD (NNSD) given in Eq. (13) for each set independently, where $a = 1, \ldots, 6$ represents the tile images in a certain set, and the $b = 1, \ldots, 7$ represents different sets of tile images, cf. Fig. 7. Total Normalized NSD (TNNSD) for each K is given in Eq. (14). Table 1 gives NNSD results for all sets where $K = 1$ represents the NNSD results for SCD method. One can observe that the NNSD decreases proportionally as K values increase. The best NNSD values, bolded in Table 1 are mostly for $K = 25$. One should take into account that there is an increase in the number of calculations as K values increase. From Eqs. (5) to (7) it is discernible that the number of calculations greatly depends on the K value. It is necessary to leverage between the number of calculations and the NNSD value because of the algorithm performance hit with larger K values. The graph in Fig. 10 depicts the cumulative TNNSD performance between SCD and MLSCD for each of the tested K. From the graph it is visible that the increase in the TNNSD performance between two adjacent K is significantly reduced after $K = 11$.

$$NNSD_{b,K} = \frac{1}{6}\sum_{a=1}^{6}\left(\frac{NSD_{a,b,K} - \min(NSD_{b,K})}{\max(NSD_{b,K}) - \min(NSD_{b,K})}\right), \tag{13}$$

$$TNNSD_K = \frac{1}{7}\sum_{b=1}^{7}\left(NNSD_{b,K}\right). \tag{14}$$

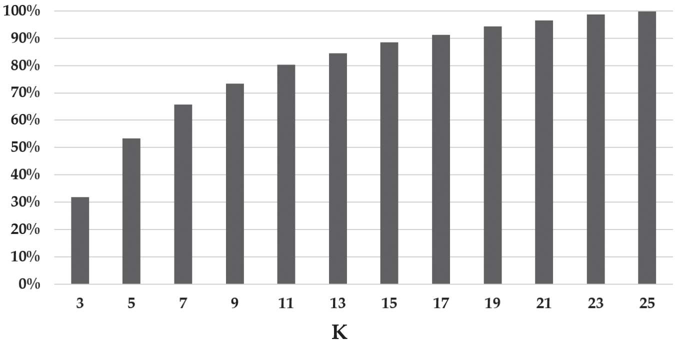


Fig. 10. Cumulative TNNSD performance between SCD and MLSCD with different K values

The second qualitative measure is the PBM measure. To additionally validate the segmentation results and give a comparison to other methods, an evaluation method from [17] is used. Authors in [17] compared the SCD-based BTS method with eight other well-known segmentation methods. Three measures are used for quantitative evaluation: misclassification error (ME), relative foreground area (RAE), and modified Hausdorff distance (MHD) [25] given in Eq. (15) to (17).

$$ME = 1 - \frac{|B_R \cap B| + |T_R \cap T|}{|B_R| + |T_R|}, \tag{15}$$

$$RAE = \frac{\big|\,|T_R| - |T|\,\big|}{\max\left\{\,|T_R|, |T|\,\right\}} \tag{16}$$

$$\begin{aligned} MHD(T_R, T) &= \max\left(d\left(T_R, T\right), d\left(T, T_R\right)\right), \\ d(A,B) &= \frac{1}{|A|}\sum_{a \in A}\min_{b \in B}\|a - b\|. \end{aligned} \tag{17}$$

In Eqs. (15) to (17), T_R is the set of tile pixels and B_R is the set of background pixels in the reference image (ground truth image). For the test biscuit tile image, B is the set of pixels classified as background, and T is the set of pixels classified as tile. The

Table 1
NNSD results

K	Bianco Mat	Mirasol	Madielle Dark	Vellagio	Botticiono	Sabuni Beige	Sabuni Mosaico	TNNSD
1	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
3	0.59012	0.73733	0.81157	0.54751	0.78311	0.67184	0.63513	0.68237
5	0.40050	0.55960	0.57556	0.29169	0.58838	0.44109	0.40922	0.46658
7	0.27195	0.38234	0.45427	0.19173	0.49724	0.30277	0.29477	0.34215
9	0.28297	0.26267	0.34562	0.14309	0.38672	0.22295	0.21630	0.26576
11	0.25850	0.20308	0.26105	0.11461	0.24292	0.14282	0.15366	0.19666
13	0.22938	0.13908	0.22753	0.09404	0.17520	0.10409	0.11198	0.15447
15	0.16793	0.10839	0.17598	0.07630	0.12943	0.07360	0.07319	0.11497
17	0.12357	0.07530	0.13308	0.05560	0.13030	0.04441	0.05472	0.0881
19	0.07722	0.04902	0.09393	0.02803	0.08803	0.02102	0.04277	0.05715
21	0.04841	0.02732	0.06834	0.01527	0.04879	0.01305	0.02542	0.03523
23	0.01970	0.00656	0.02736	0.00649	0.02511	**0.00168**	0.00997	0.01384
25	**0.00000**	**0.00099**	**0.00000**	**0.00000**	**0.00000**	0.00596	**0.00487**	**0.00169**

cardinality of the set is denoted with $|\,.\,|$, while a and b are pixels from corresponding sets A and B, respectively.

Values of the ME, RAE and MHD results are in the range $[0\dots1]$, where smaller number represents better performance [17]. Therefore, PBM is defined as

$$\mathrm{PBM} = ME + RAE + MHD. \tag{18}$$

Average ME, RAE, MHD, and PBM results for dataset are presented in Table 2. According to PBM, the best overall performance (bolded in Table 2) is for $K = 23$, where ME, RAE and MHD are best for $K = 11$, $K = 25$ and $K = 11$, respectively.

Table 2
Normalised overall results for ME, RAE, MHD and PBM

K	ME	RAE	MHD	PBM
1	0.99946	0.85290	0.78903	2.64139
3	0.83160	0.72994	0.72934	2.29088
5	0.66152	0.63575	0.6197	1.91704
7	0.48622	0.56592	0.50523	1.55738
9	0.33325	0.50683	0.40391	1.24399
11	**0.18194**	0.42095	**0.28036**	0.88325
13	0.19180	0.36865	0.29018	0.85063
15	0.21842	0.29720	0.32098	0.83660
17	0.21896	0.28999	0.31894	0.82790
19	0.22502	0.26163	0.33698	0.82363
21	0.22546	0.22521	0.34086	0.79153
23	0.23361	0.19602	0.36059	**0.79022**
25	0.24224	**0.19782**	0.36343	0.80348

The third qualitative evaluation method is a new contour angles comparison (CAC) method. The CAC evaluation is calculated with the Algorithm 2.

Algorithm 2 Contour Angles Comparison

```
procedure CAC($\Phi$, $\Phi'$)
    $\Phi_{GT} \leftarrow \Phi$, $\Phi_{MLSCD} \leftarrow \Phi'$
    $Error \leftarrow 0$
    if $length(\Phi_{GT}) \neq length(\Phi_{MLSCD})$ then
        $Error \leftarrow abs(length(\Phi_{GT}) - length(\Phi_{MLSCD}))$
        $\Phi_{MLSCD} \leftarrow RezizeVectorToMatch(\Phi_{GT})$
    end if
    for $i = 1$ to $length(\Phi_{GT})$ do
        if $\Phi_{GT}(i) \neq \Phi_{MLSCD}(i)$ then
            $Error \leftarrow Error + 1$
        end if
    end for
end procedure
```

The CAC value is the absolute difference between the ground truth contour descriptor Φ, cf. Section 3.4.2, and the one obtained with the proposed MLSCD method Φ'. The CAC values for all images in the dataset are shown in Table 3. Values $TCAC_K$ are total CAC evaluation results for all images in a single set, and for all sets in the database for a certain K. The CAC and $TCAC$ values are represented in percentages with

$$CAC_{b,k} = \left(\frac{|\Phi_{a,K} - \Phi'_{a,K}|}{|\Phi_{a,K}|}\right) \cdot 100, \tag{19}$$

$$TCAC_K = \frac{1}{7}\sum_{b=1}^{7} CAC_{b,K}\,. \tag{20}$$

Table 3
CAC relative values

K	Bianco Mat	Mirasol	Madielle Dark	Vellagio	Botticiono	Sabuni Beige	Sabuni Mosaico	TCAC
1	5.40%	6.60%	197.15%	5.48%	12.51%	23.61%	11.57%	37.47%
3	4.68%	5.64%	175.96%	4.25%	10.77%	15.54%	7.67%	32.07%
5	4.78%	4.89%	152.74%	3.83%	8.96%	9.56%	6.50%	27.32%
7	4.20%	4.41%	134.92%	3.53%	7.00%	6.04%	5.55%	23.67%
9	3.59%	4.04%	129.88%	3.36%	6.04%	5.13%	4.96%	22.43%
11	0.20%	3.75%	127.57%	3.03%	4.88%	4.50%	4.24%	21.17%
13	2.16%	3.71%	124.88%	3.23%	4.17%	4.33%	4.02%	20.93%
15	4.28%	3.65%	116.25%	3.28%	3.95%	4.27%	3.82%	19.93%
17	4.39%	3.53%	114.29%	3.27%	4.00%	4.16%	3.73%	19.63%
19	4.75%	3.46%	113.01%	3.21%	4.01%	4.11%	3.74%	19.47%
21	4.76%	3.43%	108.81%	3.16%	3.88%	4.06%	3.70%	18.83%
23	4.91%	3.30%	105.82%	3.13%	3.77%	4.03%	3.68%	18.38%
25	5.05%	3.31%	104.30%	3.06%	3.77%	4.02%	3.67%	18.17%

From *CAC* results given in Table 3, it is obvious that the segmentation of the biscuit tiles in set Madielle Dark has poor performance, while for other sets MLSCD segmentation method can be successfully applied. Similar to TNNSD performance increase, cf. Fig. 10, TCAC values show that greater K value results in better segmentation performance. After $K = 11$ performance improvement is not notable, cf. Fig. 11.

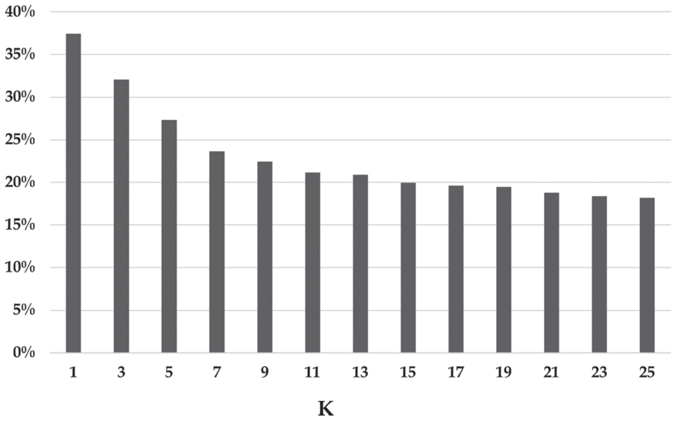


Fig. 11. TCAC performance depending on the K value

Overall results, presented in Table 4, are calculated as a sum of the three evaluation methods, *TNNSD*, *PBM* and *TCAC*, where each method is normalized in the range from 0 (lowest performance) to 1 (highest performance) for each of the tested K using

$$X_K = 1 - \frac{X_K - \min(X)}{\max(X) - \min(X)}, \tag{21}$$

where X present overall results for each evaluation method independently.

Experimental analysis shows that higher K results in better segmentation performance according to each evaluation method, except *PBM* where the best result is achieved with $K = 23$, bolded in Table 4. Nevertheless, the performance improvement is notable for K ranging from 1 to 11, while higher values of K do not significantly affect the performance. Results presented in Table 4 show that for $K = 25$ MLSCD method shows the best segmentation results.Values for $K > 25$ are not given in the experimental analysis because there is no significant increase in the overall segmentation performance. Additionally, from Table 4, it is visible that there is a small increase in the overall performance after $K = 21$.

Table 4
Overall segmentation evaluation results

K	*TNNSD*	*PBM*	*TCAC*	*Overall*
1	0.00000	0.00000	0.00000	0.00000
3	0.31763	0.18935	0.27976	0.78673
5	0.53342	0.39130	0.52584	1.45055
7	0.65785	0.58559	0.71525	1.95868
9	0.73424	0.75487	0.77928	2.26839
11	0.80334	0.94975	0.84474	2.59783
13	0.84553	0.96737	0.85704	2.66994
15	0.88503	0.97495	0.90887	2.76884
17	0.91186	0.97965	0.92452	2.81603
19	0.94285	0.98195	0.93264	2.85745
21	0.96477	0.99930	0.96584	2.92991
23	0.98616	**1.00000**	0.98918	2.97534
25	**0.99831**	0.99284	**1.00000**	**2.99115**

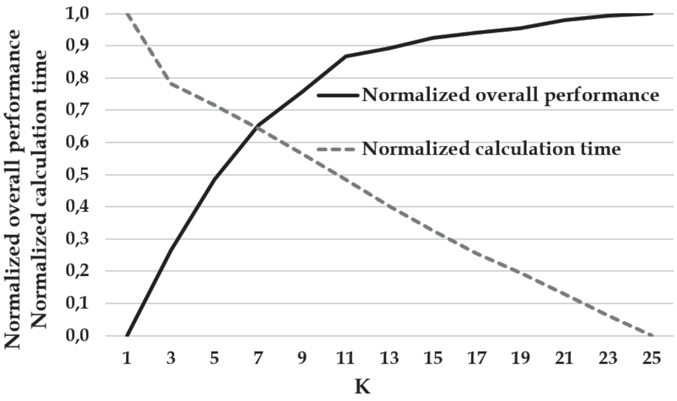


Fig. 12. Cost-effective relationship of the MLSCD

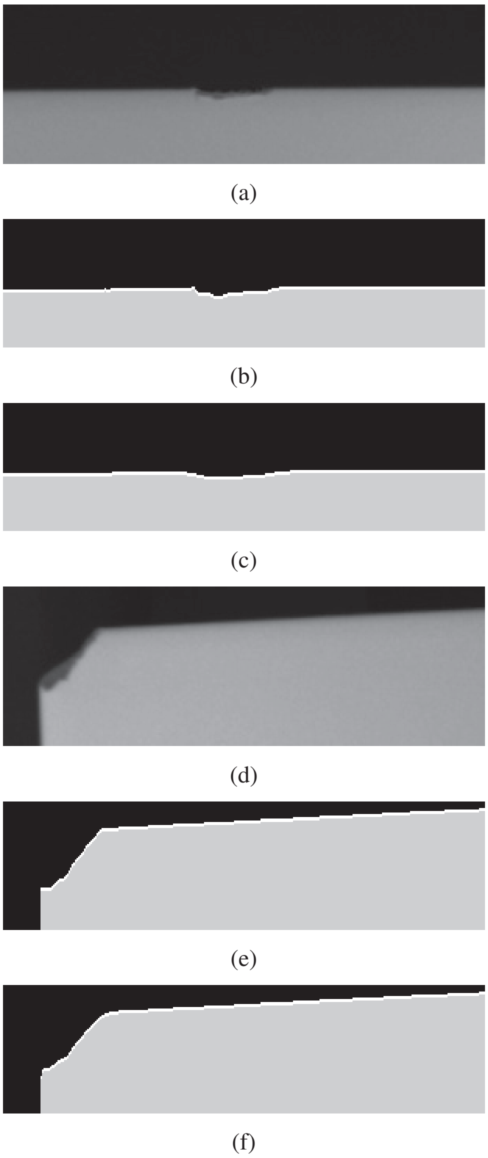


Fig. 13. Edge and corner defects example from Bianco Mat image set: input grayscale image with edge defect (a); MLSCD segmentation of the image (a) for $K = \{1, 25\}$ (b-c); input grayscale image with corner defect (d); MLSCD segmentation of the image (d) for $K = \{1, 25\}$ (e-f)

Implemented MLSCD method has linear time complexity, where K is the dependent argument. Figure 12 shows the relationship between calculation time and performance results. To present data in a more generalized form, all values are normalized. For normalized calculation time and normalized overall performance, 1 represents the best and 0 the worst possible value. All execution times for sequential MLSCD implementation are measured on a personal computer using a single CPU core. From the presented graph, it is clear that the performance and calculation time are in an inversely proportional relationship. The crossing point for the tested dataset occurs when $K = 7$, which presents the best cost-effective performance. For the sequential implementation of the MLSCD method, considering the complexity, segmentation performance, and resources, desired K should be selected.

In [17] SCD-based BTS method outperforms AEM-GC, OTSU, ACWE, RF, HMRFEM, MET, and GF segmentation methods. Therefore, from the presented segmentation evaluation results we can conclude that the MLSCD method outperforms the aforementioned methods.

In the rare case along the convey line biscuit tile can be physically damaged, which results in large edge and/or corner defects, cf. Figs. 13a and d. Currently, the available dataset has only a few images with edge and/or corner defects. Therefore, only non-defected images of biscuit tiles are considered in this work. As it is shown in Fig. 13 the MLSCD method can successfully segment defected biscuit tiles. Depending on the selected K value, the edge defect is slightly mitigated but it is not lost.

5. Conclusions

In this paper, MLSCD statistical segmentation method is presented. It is based on the application of the SCD method on K adjacent rows/columns. MLSCD method is applied to images acquired on a computer vision station prototype for biscuit tile to segment biscuit tiles from their background. Captured images are saved as *.jpg* with a maximum resolution of 2592×2048 pixels. Dataset is formed with seven different image sets, where each set contains six images.

Experimental results are evaluated using three different evaluation methods, *NSD*, *PBM*, and *CAC*. *NSD* is the number of edge pixel jumps on the segmented images, *PBM* is pixel-based statistical evaluation measure and *CAC* is the contour evaluation measure. Results show a significant performance increase of the MLCSD method compared to SCD where an increase of adjacent rows/columns results in greater segmentation performance. Best overall results were obtained using the highest selected K, in our case $K = 25$. The complexity of the calculation is proportional to performance, so a cost-effective analysis should be considered.

MLSCD method can be successfully applied to various biscuit tile designs without using the image preprocessing step. Only exception are high contrast designs where additional preprocessing is necessary. Future work will include implementing a preprocessing step in the presented MLSCD method for the possible increase of segmentation performance on high contrast designs. Moreover, additional image sets are required to test the proposed method and show invariance to any type of biscuit tile design.

Acknowledgements. This work was supported in part by the Interreg IPA Cross-border Cooperation Programme Croatia-Serbia 2014–2020 (project acronym SeNsWetlands, project no. 2017HR-RS135).

References

[1] G.M.A. Rahaman and M. Hossain, "Automatic Defect Detection and Classification Technique from Image: A Special Case Using Ceramic", *Int. J. Comput. Sci Inf. Secur*. 1(1), 9 (2009).

[2] M. Villalon-Hernandez, D. Almanza-Ojeda, and M. Ibarra-Manzano, "Color-Texture Image Analysis for Automatic Failure Detection in Tiles", in *Pattern Recognition*, MCPR 2017. *Lecture Notes in Computer Science*, vol. 10267, pp. 159–168, eds. J.A. Carrasco-Ochoa, J.F. Martínez-Trinidad, and J.A. Olvera-López, Springer International Publishing, Cham, 2017.

[3] M.H. Karimi and D. Asemani, "Surface defect detection in tiling industries using digital image processing methods: Analysis and evaluation", *ISA Trans*. 53(3), 834–844 (2014).

[4] F.S. Najafabadi and H. Pourghassem, "Corner defect detection based on dot product in ceramic tile images", In *2011 IEEE 7th International Colloquium on Signal Processing and its Applications*, Penang, Malaysia, 2011, pp. 293–297.

[5] R. Mishra, C.L. Chandrakar, and R. Mishra, "Surface Defects Detection for Cermaic Tiles Using Image Processing and Morphological Techniques", *Appl. Sci*. 2(2), 17 (2012).

[6] T. Czimmermann *et al*., "Visual-Based Defect Detection and Classification Approaches for Industrial Applications – A SURVEY", *Sensors* 20(5), 1459, (2020).

[7] V. Lebedev and V. Lempitsky, "Speeding-up convolutional neural networks: A survey", *Bull. Pol. Acad. Sci. Tech. Sci*. 66(6), 799–810 (2018).

[8] N. Wang, X. Zhao, Z. Zou, P. Zhao, and F. Qi, "Autonomous damage segmentation and measurement of glazed tiles in historic buildings via deep learning", *Comput.-Aided Civil Infrastruct. Eng*. 35(3), 277–291 (2020).

[9] X. Gu and Y. Sun, "Image analysis of ceramic burning based on cellular automata", *EURASIP J. Image Video Process*. 2018(1), 110 (2018).

[10] T. Matić, I. Vidović, and Ž. Hocenski, "Real Time Contour Based Ceramic Tile Edge and Corner Defects Detection", *Teh. Vjesn.-Technical Gazette* 20(6), 8 (2013).

[11] Ž. Hocenski, T. Keser, and A. Baumgartner, "A Simple and Efficient Method for Ceramic Tile Surface Defects Detection", In *2007 IEEE International Symposium on Industrial Electronics*, Vigo, Spain, 2007, pp. 1606–1611.

[12] T. Matić, I. Aleksi, and Ž. Hocenski, "CPU, GPU and FPGA Implementations of MALD: Ceramic Tile Surface Defects Detection Algorithm", *Automatika* 55(1), 9–21 (2014).

[13] J. Zhuang, L. Yang, and J. Li, "An Improved Segmentation Algorithm Based on Superpixel for Typical Industrial Applications", In *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*, Hangzhou, China, 2018, pp. 366–370.

[14] X. Zhou, Y. Wang, Q. Zhu, J. Mao, C. Xiao, X. Lu, and H. Zhang, "A Surface Defect Detection Framework for Glass Bottle Bottom Using Visual Attention Model and Wavelet Transform", *IEEE Trans. Ind. Inform*. 16(4), 2189–2201 (2020).

[15] X. Yan, L. Wen, and L. Gao, "A Fast and Effective Image Preprocessing Method for Hot Round Steel Surface", *Math. Probl. Eng*., 2019, 1–14 (2019).

[16] R. Cunha *et al*., Applying Non-destructive Testing and Machine Learning to Ceramic Tile Quality Control", In *2018 VIII Brazilian Symposium on Computing Systems Engineering (SBESC)*, Salvador, Brazil, 2018, pp. 54–61.

[17] T. Matić, I. Aleksi, Ž. Hocenski, and D. Kraus, "Real-time biscuit tile image segmentation method based on edge detection", *ISA Transactions* 76, 246–254 (2018).

[18] S.M. Kay, *Fundamentals of Statistical Signal Processing: Practical Algorithm Development*, Prentice-Hall PTR, 2013.

[19] L.G. Shapiro and R.M. Haralick, Computer and Robot Vision, Addison-Wesley Publishing Company, 1992.

[20] Z. Hocenski and T. Keser, "Failure detection and isolation in ceramic tile edges based on contour descriptor analysis", In *2007 Mediterranean Conference on Control & Automation*, Athens, Greece, 2007, pp. 1–6.

[21] R.C. Gonzalez and R.E.Woods, *Digital Image Processing*, Pearson, 2018.

[22] Ž. Hocenski, T. Matić, and I. Vidović, "Technology transfer of computer vision defect detection to ceramic tiles industry", In *2016 International Conference on Smart Systems and Technologies (SST)*, Osijek, Croatia, 2016, pp. 301–305.

[23] M. Montorsi, C. Mugoni, A. Passalacqua, A. Annovi, F. Marani, L. Fossa, R. Capitani, and T. Manfredini, "Improvement of color quality and reduction of defects in the ink jet-printing technology for ceramic tiles production: A design of experiments study", *Ceram. Int*. 42(1, Part B), 1459–1469 (2016).

[24] The GIMP Development Team. GIMP, 2019.

[25] A.Z. Arifin and A. Asano, "Image Segmentation by Histogram Thresholding Using Hierarchical Cluster Analysis", *Pattern Recognit. Lett*., 27(13), 1515–1521 (2006).