1. Marcin POŁOMSKI[1,2], 2. Tomasz KRASZEWSKI[1], 3. Artur PASIERBEK[1,2], 4, Anna PIWOWAR[1,3]

Centrum Badawczo – Rozwojowe GLOKOR Sp. z o.o. (1),
Politechnika Śląska, Wydział Automatyki, Elektroniki i Informatyki (2),
Politechnika Śląska, Wydział Elektryczny (3),
ORCID: 1. 0000-0001-8785-2377, 2.0000-0003-2208-85362. 3. 0000-0002-2137-4031, 4. 000-0002-5578-3757

doi:10.15199/48.2024.01.47

Feature extraction method for classification in workpieces defects

Streszczenie. *Artykuł stanowi opis metody ekstrakcji cech wykorzystanej w klasyfikacji wad detali poddawanych obróbce. Praca stanowi część prac związanych z opracowaniem linii do automatycznej identyfikacji wad na elementach obrabianych. W artykule dokonano opisu metody wyodrębniania cech, w której zastosowano wyniki wokselizacji do otrzymania parametrów charakterystycznych wad. Wyniki zilustrowano przykładami.*

Abstract*. The article is a description of the feature extraction method, used in the classification of defects in workpieces undergoing a process. The publication is a part of the work related to the development of a line for automatic identification of defects on workpieces. It describes a method of extracting features, in which voxelization results were used to obtain the parameters of characteristic defects. The results are illustrated with examples (**Metody ekstrakcji cech do klasyfikacji wad detali podawanych obróbce**)*

Słowa kluczowe: detekcja wad, systemy wizyjne, ekstrakcja wad, klasyfikacja wad
Keywords: defect detection, vision systems, feature extraction, feature classification.

Introduction

The article is related to the project of research and development works aimed at determining an innovative solution in the form of a line facilitating the automatic identification of defects in elements subjected to the blast cleaning process. It can be considered an introduction to the development of vision technology for defect detection, which is one of the manufacturing steps. The vision system aims to visualize the physical characteristics of tested items enabling their measurement and quality control. Based on the data collected by the system, an automatic decision is made to proceed to the next stage of the process with precise information on the location of the defect and its characteristics, leading to the selection of an appropriate method for its removal. It is assumed that the system will consist of vision devices and an image acquisition and processing system, combining the acquisition process, image processing and software, which makes it possible to automate the stages of the control, measurement and production process [1]. Vision systems which perform quality control are used in the production process and warehouse service. The obtained image is compared to the stored patterns and deviation values. This work is a description of the feature extraction method, used in the classification of detail defects, which will be further processed.

Test model defect classification

The test model for the defect identification system was a test sample with a series of marked defects. For this purpose, a catalogue of defects, containing exemplary types of defects, was proposed:

- linear – defects which meet the length condition (1).

The geometry of a line defect has been presented in fig. 1.

(1) $b/a > 5$ and $a \leq 2$ mm

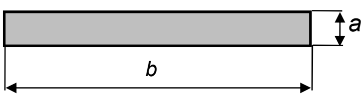


Fig. 1. Linear defect geometry.

(2) $a/b \geq 0{,}8$ and $b \leq 2\ mm$

- point – defects with measurements meeting condition (2). The geometry of a linear defect has been presented in fig. 2.

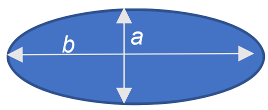


Fig. 2. Point defect geometry.

- surface – all the remaining defects, which do not meet the previously mentioned criteria.

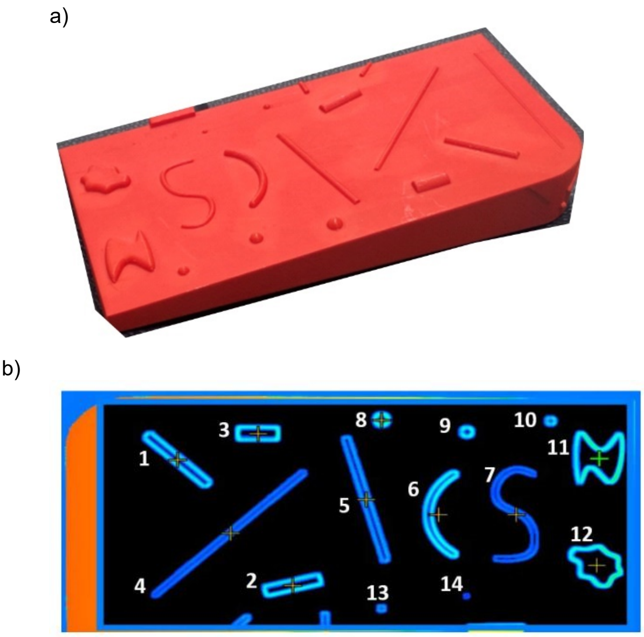


Fig. 3. Test sample a) model of the sample, b) view of the upper surface of the sample

These listed defects were generated and applied to the so-called test sample, which is an element characterized by a structure with sharp edges, without small elements in its structure (fig. 3).

For the several expected types of defects, taking into account the listed criteria, the following classification has been created: (contrasted in fig. 3b)

- linear defects: 1, 2, 4, 5, 6, 7,
- point defects: 13, 14,
- surface defects: 3, 8, 9, 10, 11, 12.

In addition, it was assumed that the height of the defect may reach a certain limit value (by default 5mm), beyond which the defect will not be subject to further processing. Furthermore, the need to determine the volume of the defect, in order to develop a machining program, was indicated.

The basic classification method assumes the occurrence of solely several types of defects, whereas, in real conditions, which was observed when analyzing scans of real objects, defects can definitely take different shapes or be a combination of many simplified variants. Another issue is the assumption that the projection of the defect on the plane ought to be analysed, which cannot precisely reflect a three-dimensional model. Such a solution may distort the actual image of the defect. The possibility of extracting a representative set of defect features was indicated, which might be considered a base for a classification useful for generalizing the decision-making process.

The proposed method also allows for a selective set of features to feed a dedicated classification algorithm, e.g. built using a decision tree (for experimentally selected boundary parameters) or another algorithm (e.g. machine learning with supervision).

In the first stage, the following defect parameters, determined during its analysis, were distinguished, constituting a set of characteristics of the classification process:

- surface cuboid, with:
  - coordinates:
    - `bbox-xmin` – minimum coordinate value *x*
    - `bbox-xmax` – maximum coordinate value *x*,
    - `bbox-ymin` - minimum coordinate value *y*,
    - `bbox-ymax` - maximum coordinate value *y*
    - `bbox-zmin` - minimum coordinate value *z*,
    - `bbox-zmax` - maximum coordinate value *z*,
  - side lengths:
    - `bbox-xsize` – on the *X* axis,
    - `bbox-ysize` - on the *Y* axis,
    - `bbox-zsize` - on the *Z* axis,
- volume of surrounding cuboid (`bbox-volume`),
- number of voxel of the defect (voxel is a cube with a defined edge length) (`num-points`),
- defect volume (`defect-volume`),
- the ratio of the volume of the defect to the volume of the surrounding cuboid (`dv-to-bbv`),
- number of voxels of the layer adjacent to the model (`num-points-adj`),
- surface area of the adjacent layer (`adjl-surface-area`),
- maximum height of the defect (`height-max`),
- average height of the defect (`avg-vec-len`).

To determine the above mentioned features, a grid of a single defect is used. The process of building a voxel grid inside the detected allowance, hereinafter referred to as voxelization [2] [3], [4] is necessary to determine the selected characteristics and related derived features.

In the first step, points describing the voxelized form of the defect are determined - to determine the parameters (and related derived parameters such as the number of voxels of the defect, the volume of the defect and the number of voxels of the layer adjacent to the model. The grid of the reference model (also used in the process of determining the characteristic parameters), which, together with the points of the voxelized defect model, allows to determine parameters such as the maximum and average height of the defect.

Thanks to the above actions, it is possible to extract the defect parameters, both during the defect voxelization process and after its completion, having the appropriate set of data. The first described scenario seems to be more efficient since the parameter extraction procedure can be incorporated into the voxelization algorithm. The method and results of obtaining defect parameters were presented in one of the sample defects of the test model.

Determination of defect parameters based on the surrounding cuboid

Examples only include key pieces of pseudo-code, allowing to understand the idea behind the parameter determination method. In the first stage, the parameters of the surrounding cuboid and the parameters resulting from its geometry are determined (fig. 4).

```
def getBoundingBoxPoints(object_name):
    obj = bpy.data.objects[object_name];
    bbox_arr = np.array(obj.bound_box);
    return bbox_arr


def getBoundingBoxBoundaries(object_name):
    obj = bpy.data.objects[object_name];
    bbox_arr = np.array(obj.bound_box);
    xmin = np.min(bbox_arr[:,0]); xmax = np.max(bbox_arr[:,0]);
    ymin = np.min(bbox_arr[:,1]); ymax = np.max(bbox_arr[:,1]);
    zmin = np.min(bbox_arr[:,2]); zmax = np.max(bbox_arr[:,2]);
    result = [(xmin, xmax), (ymin, ymax), (zmin, zmax)];
    return result

# ...
```

```
bbox_points = getBoundingBoxPoints(defect_object_name);
bbox_bounds = getBoundingBoxBoundaries(defect_object_name);
xmin = bbox_bounds[0][0]; xmax = bbox_bounds[0][1];
ymin = bbox_bounds[1][0]; ymax = bbox_bounds[1][1];
zmin = bbox_bounds[2][0]; zmax = bbox_bounds[2][1];
```

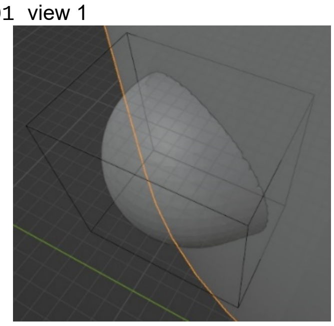
Fig. 4. Code fragment illustrating the idea of determining parameters based on the surrounding cuboid

Based on the coordinates of the surrounding cuboid, the following model parameters are calculated, fig. 5:
`xmin, xmax, ymin, ymax, zmin, zmax`, `bbox-xsize, bbox-ysize, bbox-zsize, bbox-volume`

```
features['bbox'] = {'xmin' : xmin, 'xmax' : xmax,
    'ymin' : ymin, 'ymax' : ymax,
    'zmin' : zmin, 'zmax' : zmax,
    'bbox-xsize': xmax-xmin,
    'bbox-ysize': ymax-ymin,
    'bbox-zsize': zmax-zmin,
    'bbox-volume': (xmax-xmin)*(ymax-ymin)*(zmax-zmin) };
```

Fig. 5. Code fragment illustrating parameter calculation

a) `defect_01` view 1



b). `defect_01` view 2

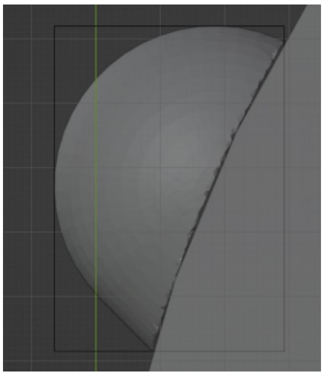


Fig. 6. Set parameters based on the surrounding cuboid

For an exemplary defect, defined `defect_01`, the result shown in fig. 6 is obtained. The obtained parameter values are summarized in table 1.
The described method was used to determine the parameters of all defects applied to the tested element.

Tab. 1. Parameter values for defect `defect_01`

parameter	value
bbox-xmin	-0,6425
bbox-xmax	2,9239
bbox-ymin	84,1475
bbox-ymax	89,1974
bbox-zmin	8,7837
bbox-zmax	13,7743
bbox-xsize	3,5664
bbox-ysize	5,0499
bbox-zsize	4,9906
bbox-volume	89,8813

```
dist_max = 0; dist_max_loca_p1 = 0; dist_max_loca_p2 = 0;
sum_dist = 0;

const color_adjacent_layer = 3; # default
counter = 0; color_index_count = 0;

# average vector components
xavg = 0; yavg = 0; zavg = 0;

for point in voxelized_model_points:
    # point has (x, y, z, c)
    # c - color value determines if voxel touches the main model
    verts.append(point);
    # find nearest point on base ideal mesh model
    res = main_bvh.find_nearest(point);
    loca = res[0]; #vector location
    norm = res[1]; #vector normal
    indx = res[2]; #int index
    dist = res[3]; #distance

    xavg = xavg + (point[0] - loca.x);
    yavg = yavg + (point[1] - loca.y);
    zavg = zavg + (point[2] - loca.z);
    sum_dist = sum_dist + dist;

    if point.c == color_adjacent_layer
        color_index_count = color_index_count + 1;

    # compute dist max
    if counter == 0:
        dist_max = dist
        dist_max_loca_p1 = loca
        dist_max_loca_p2 = (point[0], point[1], point[2])
    else:
        if dist > dist_max:
            dist_max = dist
            dist_max_loca_p1 = loca
            dist_max_loca_p2 = (point[0], point[1], point[2])

    counter = counter + 1;
```

Fig. 7. Code fragment illustrating the idea of determining the parameters based on the points of the voxelized model

Determination of parameters based on the voxelized model points

In the next step, defect parameters are determined, taking into account the set of points of the voxelized model. For this purpose, the previously indicated sets of input data are used. The process of obtaining parameters determined on the basis of points of the voxelized model has been illustrated in the pseudo-code fragment shown in Fig. 7
Based on the previously obtained results, it is possible to calculate the following parameters:

`num-points` – number of defect voxels,
`defect-volume` – volume of the defect,
`dv-to-bbv` - ratio of the volume of the defect to the volume of the surrounding cuboid,
`num-points-adj` - number of voxels of the layer adjacent to the model,
`adjl-surface-area` - surface area of the adjacent layer,
`height-max` – maximum height of the defect,
`avg-vec-len` – so-called average height of the defect.

The idea of determining these parameters is illustrated in the code fragment shown in Fig. 8

```
features['num-points'] = len(voxelized_model_points);
features['defect-volume'] = len(voxelized_model_points) * d*d*d;

bbv = features['bbox']['bbox-volume'];
dv = features['defect-volume'];
features['dv-to-bbv'] = dv/bbv;

features['num-points-adj'] = color_index_count;
features['adjl-surface-area'] = color_index_count * d*d;

features['height-max'] = dist_max;
features['avg-vec-len'] = mu.Vector((xavg, yavg, zavg)).length /
counter;
```

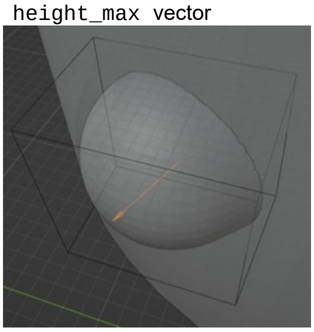
Fig. 8. Code fragment calculating the above-mentioned parameters.

At this step, for the exemplary `defect_01`, the results presented in Fig. 9 were obtained, and their values are presented in Table 2.

Tab. 1. Parameter values for `defect_01`

parameter	value
num-points	32 462
defect-volume	32,4620
dv-to-bbv	0,3612
num-points-adj	1 600
adjl-surface-area	16,0000
height-max	2,4495
avg-vec-len	0,9340

a) `defect_01 height_max` vector



b) `defect_01 avg-vec` vector

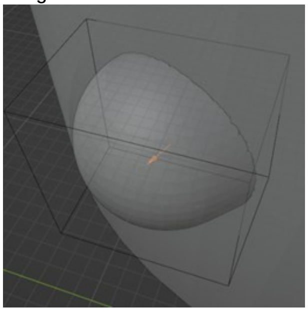
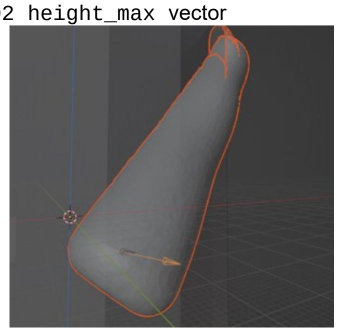


Fig. 9. Parameters determined on the basis of points of the voxelized model for `defect_01`

a) `defect_02 height_max` vector



b) `defect_02 avg-vec` vector

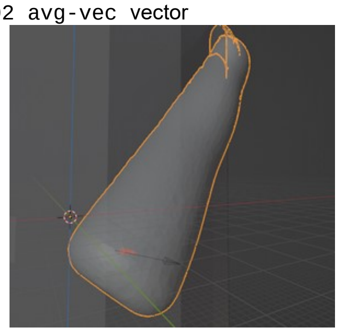
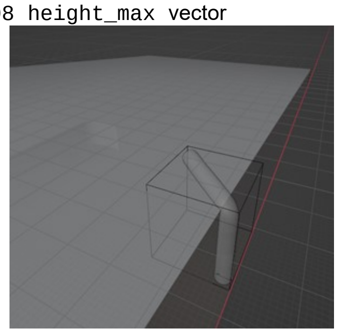


Fig. 10. Parameters determined on the basis of points of the voxelized model for `defect_02`

a) `defect_08 height_max` vector



b) `defect_08 avg-vec` vector

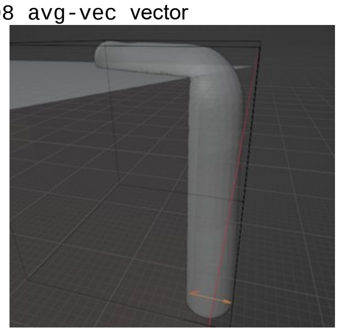


Fig. 11. Parameters determined on the basis of points of the voxelized model for `defect_08`

The described general method of determining parameters was used for each of the defects of the test model. Selected results have been visualized in the pictures below (in particular, the surrounding cuboid and vector sizes describing the maximum and the average height of the defect).

Conclusions

From the point of view of the effectiveness of the later applied classification method, appropriate selection of features in terms of suitability for further use in the classification process is considered a significant issue. The article describes the evolution of the adopted method, allowing for a more precise quantitative description of defects, followed by the extraction of characteristics which are a linear or non-linear combination of original parameters. The implemented method enables its integration into the defect extraction process, which can directly affect the performance and be significant for the effectiveness of the entire defect detection and removal process. In further research, it is possible to attempt to evaluate the usefulness of the extracted characteristics in correlation with the applied classification method.

Acknowledgements

Research financed within the framework of the project „Prace badawczo-rozwojowe zmierzające do opracowania linii do automatycznej identyfikacji wad na elementach obrabianych z opracowaniem technologii ich usuwania strumieniową obróbką ścierną na stanowiskach zrobotyzowanych w warunkach przemysłowych", No.: POIR.04.01.04-00-0083/20

***Autorzy**: dr inż. Tomasz Kraszewski, Centrum Badawczo – Rozwojowe GLOKOR Sp. z o.o., ul. Górnych Wałów 27A, 44-100 Gliwice t.kraszewski@glokor.eu, dr inż. Artur Pasierbek, dr inż. Marcin Połomski, Politechnika Śląska, Wydział Automatyki Elektroniki i Informatyki ul. Akademicka 16, 44-100 Gliwice e-mails: artur.pasierbek@polsl.pl, marcin.połomski@polsl.pl, dr inż. Anna Piwowar, Politechnika Śląska, Wydział Elektryczny, ul. Akademicka 10, 44-100 Gliwice, e-mail: anna.piwowar@polsl.pl.*

REFERENCES

[1]. Kraszewski T., Piwowar A., Frączek R., Świszcz P., Bis Ł. Visual defect detection methods overview, *Przegląd Elektrotechniczny*, 98 (2022), No 11, pp. 269-272.

[2]. Huang J., Yagel R., Filippov V., Kurzion Y.: An accurate method for voxelizing polygon meshes, EEE *Symposium on Volume Visualization*. (1998)119-126. 10.1109/SVV.1998.729593.

[3]. Nourian P., Goncalves R., Zlatanova S., Ohori K., Vo A.: Voxelization Algorithms for Geospatial App., *Methods X*. 3 (2016) 10.1016/j.mex.2016.01.001.

[4]. Tarbutton J., Kurfess T., Tucker T.: Graphics Based Path Planning for Multi-Axis Machine Tools, *Computer-Aided Design and Applications*. 7. 835-845. 10.3722/cadaps.2010.835-845 (2013).