The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLVIII-2/W2-2022
Optical 3D Metrology (O3DM), 15–16 December 2022, Würzburg, Germany

USING MACHINE LEARNING ON DEPTH MAPS AND IMAGES FOR TUNNEL EQUIPMENT SURVEYING

Florian Barcet, Maxime Tual, Philippe Foucher*, Pierre Charbonnier

Cerema, Project-Team ENDSUM, Strasbourg, France
(florian.barcet, maxime.tual, philippe.foucher, pierre.charbonnier)@cerema.fr

Technical Commission II

KEY WORDS: Tunnel surveying, Laser Scanning, Mobile acquisition, Documentation, Building Information Modeling

ABSTRACT:

In this contribution, we illustrate how to use high-resolution 3D data and images to model the global shape of a tunnel and to survey its equipment, in a semi-automatic way using pattern recognition and machine learning techniques. We first implement a robust B-spline fitting algorithm, based on a parametric family of M-estimators that allows an efficient deterministic optimization strategy, to accurately model the tunnel lining using range data, despite the presence of acquisition artifacts and significant perturbations related to the equipment and some surface defects. The residual maps from the robust fit can be exploited to segment the equipment in an unsupervised manner using clustering algorithms, but at the cost of post-processing which makes the method rather ineffective for routine use. However, we deploy it to annotate data for the supervised learning of a *deep learning* model, namely Mask R-CNN. We comment on the first results, obtained on a still limited number of examples, and from image data only, and discuss the possibilities of improving the method, in the immediate or longer term.

1. INTRODUCTION

The survey of existing tunnels can have several purposes, such as documentation and BIM (Building Information Modeling) or safety diagnostics. This task is often costly and error-prone when performed on-site. Nowadays, sensors can provide high-resolution 3D data (e.g. in the form of depth maps) and images. In this paper, we illustrate how pattern recognition and machine learning techniques can be used to exploit this data, with the aim of performing tunnel surveys off-line, as automatically as possible.

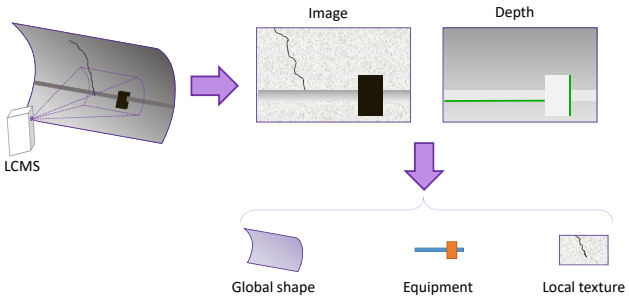

Figure 1. Outline of the approach taken

In (Foucher et al., 2019), we showed how it was possible to adapt a high-resolution sensor, namely Pavemetric's LCMS (Laser Crack Measurement System), initially designed for road surface inspection (Laurent and Hébert, 2002), to tunnel data surveying by equipping a vehicle with a carrying platform configurable in height and angle. The LCMS sensor analyzes the projection of a laser line on the tunnel lining, providing at each acquisition a depth profile and an intensity profile. These are progressively aggregated as the vehicle moves forward to form images and depth maps, with a spatial resolution of 1×2 mm and a sub-millimeter depth resolution. As for any laser-based sensor, the LCMS measurement can be disturbed by optical phenomena, occlusions or defocus. When identified by the sensor, the resulting invalid / out-of-range (IOoR) values are indicated in the depth image by an arbitrary constant (depicted as green pixels in the diagram of Fig. 1).

Depth maps include information at different scales: a global envelope that represents the general geometry of the tunnel (metric scale), foreground objects that often correspond to its equipment (decimetric to centimetric scale), and the local texture of the lining, whose alterations may be related to the presence of defects or structural elements such as joints (centimetric to millimetric or sub-millimetric scale). Moreover, images convey appearance information, which can also be valuable to detect objects of interest, or defects.

In this contribution, we focus on developing methods to extract objects at these different scales (see Fig. 1), from image and/or depth data. More specifically, we are interested in modeling the general shape of tunnels and in segmenting their equipment, leaving defect detection outside the scope of this paper. The latter topic, especially crack, water leakage and exposed iron detection from images is receiving considerable attention in the recent years and we refer the reader to e.g. (Gupta and Dixit, 2022) or (Nguyen et al., 2022) for a recent overview. Note that works explicitly using depth information for crack detection (Gui et al., 2019) are still rather rare in the literature.

Modeling the shape of tunnels, mainly for profilometry purposes, is a relatively classical task. However, the appearance of robust methods, allowing to adjust composite surfaces in the presence of atypical data, or outliers, is quite recent. In the spirit of (Xu and Yang, 2020), we first implement a robust B-spline fitting algorithm to accurately model the tunnel surface using range data, despite acquisition artifacts and significant perturbations related to equipment or to surface defects. The contribution here is that we rely on a parametric family of M-estimators which allows an efficient deterministic optimization strategy,

* Corresponding author

and an increased robustness. The proposed method is described in Section 2 of this paper.

The second contribution of the paper relates to the segmentation of equipment. Apart from paper (Xu et al., 2021), where some examples of equipment appear as "interferences" in the context of defect detection, we are not aware of any work specifically involving equipment in tunnels. Two techniques are explored here. A first, unsupervised approach uses the residuals of the robust fit, i.e. a version of the depth map corrected from the global shape of the tunnel, to identify equipment. It is described in Section 3. A second approach, supervised, is based on a deep learning model (namely Mask R-CNN). It operates on image data, while its learning is performed from data labeled with the previous method (hence, using range data). It is described in Section 4.

Finally, in Section 5, we conclude the paper and consider ways to improve the proposed methods.

We note that a preliminary and abridged version of the first two sections of this paper appeared in (Tual et al., 2021).

2. ROBUST DEPTH MAP MODELING

In the first place, we model the global shape of the tunnel by fitting a surface on the depth data $\{x_k, y_k, d_k\}_{k\in[1,K]}$. To this end, we use a B-spline model, as in (Xu and Yang, 2020). It is a surface $\mathcal{M}$ parameterized by $(u, v) \in [0, 1] \times [0, 1]$, defined as a combination of $n \times m$ polynomial basis functions :

$$\mathcal{M}(u,v) = \sum_{i=1}^{n}\sum_{j=1}^{m} \mathbf{N}_i(u)\mathbf{N}_j(v)\boldsymbol{\beta}_{j+m(i-1)} \tag{1}$$

where $\mathbf{N}_i$ and $\mathbf{N}_j$ are cubic spline functions. These are defined from a nodal vector using the recursive Cox - de Boor formula, see (Piegl and Tiller, 2012) for details. The placement of the nodes points allows to finely control the local level of description of the surface: distant nodes provide a coarse representation, while close nodes allow a finer one. Since duplicating a node decreases by one the level of continuity of the curve, it is even possible to model discontinuities.

In the discrete setting, noting $\mathbf{M} = [\mathbf{x}, \mathbf{y}, \mathbf{d}]$, the $(K \times 3)$ matrix of depth data, eq. (1) can be written as $\mathbf{M} = \mathbf{X}\boldsymbol{\beta}$, where $\mathbf{X}$ is the so-called *design matrix* and $\boldsymbol{\beta}$ is a $(nm \times 3)$ matrix of *control points*. Each row k of the design matrix contains all combinations of the basis functions evaluated at point $(\mathbf{u}_k, \mathbf{v}_k)$. Each column contains the 2D spline corresponding to the control point $\boldsymbol{\beta}_{j+m(i-1)}$. For more details on the construction of the design matrix, we refer the reader to (Piegl and Tiller, 2012).

$$\mathbf{X} = \begin{bmatrix} (\mathbf{N}_i(\mathbf{u}_1)\mathbf{N}_j(\mathbf{v}_1))_{(i,j)\in[1,n]\times[1,m]} \\ \vdots \\ (\mathbf{N}_i(\mathbf{u}_k)\mathbf{N}_j(\mathbf{v}_k))_{(i,j)\in[1,n]\times[1,m]} \\ \vdots \end{bmatrix} \tag{2}$$

The estimation of the unknown vector $\boldsymbol{\beta}$ can be done in a least squares (LS) sense, i.e. by minimizing the sum of squared residuals: $J_{LS} = \sum_k r_k^2$, where the k-th residual is defined as

$$r_k = \|M_k - X_{k,.}\boldsymbol{\beta}\|_2. \tag{3}$$

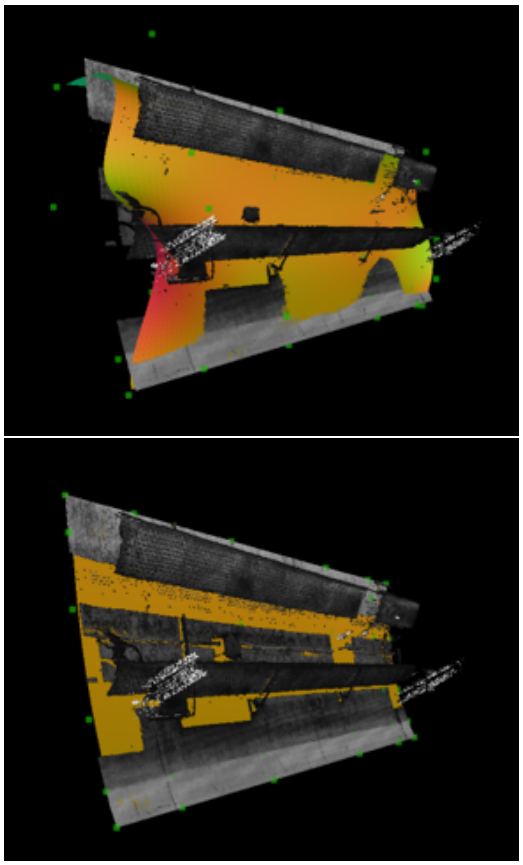

Figure 2. B-spline tunnel surface modeling with LS (top) or robust (bottom) fit. The adjusted surface is shown in brown, superimposed on the 3D data, green squares are control points

However, since LS are sensitive to erroneous data, IOoR points as well as foreground objects and artifacts tend to attract the surface and distort it unduly (see the example of LS fit in Fig. 2, top). To prevent this, we use a robust estimation technique, namely M-estimators, which minimize the criterion:

$$J_M(\boldsymbol{\beta}) = \sum_{k=1}^{k=K} \rho(r_k), \tag{4}$$

where the so-called potential function, ρ, is chosen in the Smooth Exponential Family (Tarel et al., 2002):

$$\rho(r) = \begin{cases} \frac{1}{\alpha}\left[(1+r^2)^\alpha - 1\right] & \alpha \neq 0 \\ \ln(1+r^2) & \alpha = 0 \end{cases} \tag{5}$$

where α is a parameter that controls the shape of ρ (see Fig. 3, top). Taking $\alpha = 1$, one obtains the quadratic (non-robust) function, while $\alpha = 0.5$ leads to the quasi-Laplace potential, which is close to the Huber potential. Smaller α values lead to non-convex potentials, whose influence functions (Hampel et al., 1986) are redescending, providing enhanced robustness to outliers as α decreases. For $\alpha = 0$, one obtains the t-Student function, used in (Yang et al., 2019, Xu and Yang, 2020). The $alpha = 0$ case corresponds to the Geman-McClure function (Geman and McClure, 1985), which has a horizontal asymptote and is particularly tolerant to outliers. In order to avoid the local minima problems related to the optimization of non-convex criteria, it is strongly advisable to use them in a *continuation* approach, i.e. decreasing α gradually, starting each

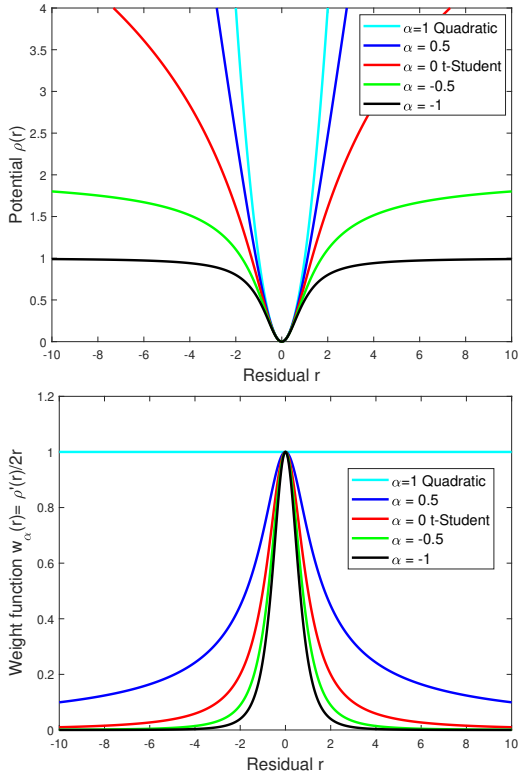

Figure 3. SEF potential (top) and weight (bottom) functions, as introduced in (Tarel et al., 2002)

time the optimization from the solution found with the previous value. To perform the optimization of criterion (4), which leads to nonlinear normal equations, we use the half-quadratic theory, which defines an *augmented* loss function with the same minimum as J_M:

$$J_{WLS}(\boldsymbol{\beta}, w) = \sum_k w_k r_k^2 + \xi(w_k). \tag{6}$$

The augmented problem has two unknowns, but when $\boldsymbol{\beta}$ is fixed, the minimizer of J_{WLS} is given by the so-called weight function: $w_k = w_\alpha(r_k) = \rho'(r_k)/2r_k, \forall k \in [1, K]$ and when w is fixed, minimizing J_{WLS} with respect to $\boldsymbol{\beta}$ is a weighted LS problem, which can be straightforwardly solved. In these conditions, it is wise to use a strategy of alternate optimizations w.r.t. each variable, which leads to an iteratively reweighted least squares (IRLS) algorithm (Holland and Welsch, 1977). As can be seen from the shape of the weighting function (Fig. 3, bottom), strong residuals, related to outliers, receive a low weight, the more so as α is low, while small ones have a weight close to one. In this framework, it is easy to cancel the weight of the IOoR points, using an indicator vector, $\mathbf{o}$. Finally, a factor σ is introduced to account for the scale of residuals. The whole process is described in Algorithm 1. As shown on Fig. 2 (bottom), robust estimation is almost insensitive to the presence of cable trays, lighting and artifacts. The remaining brown areas mostly correspond to parts of the surface occluded by equipment.

Several practical details need to be taken into account when implementing this algorithm. The Spline basis functions have a rather limited spread, so the design matrix contains a large majority of zeros. This is why it is advantageous to manipulate it with sparse arithmetic. On-the-fly code generation with the Python library Numba (Lam et al., 2015), allows to significantly accelerate calculations. Using the properties of multiplication by diagonal matrices also makes the calculation of $\hat{\boldsymbol{\beta}}_{WLS}$ lighter. Finally, sub-sampling the data allows to reduce the memory and computational resources used.

Input: Vector of α values: $\boldsymbol{\alpha}$;
Scale parameter: σ ;
Matrix of observations: $\mathbf{M}$;
IOoR indicator vector: $\mathbf{o}$;
Design matrix: $\mathbf{X}$;
Output: Estimated control points: $\hat{\boldsymbol{\beta}}$
Initialization $\hat{\boldsymbol{\beta}}^{(0)} \leftarrow \hat{\boldsymbol{\beta}}_{LS} = \left(\mathbf{X}^\mathsf{T}\mathbf{X}\right)^{-1}\mathbf{X}^\mathsf{T}\mathbf{M}$;
$t \leftarrow 0$;
foreach $\boldsymbol{\alpha}$ **do**
  $i \leftarrow 0$;
  repeat
    $\mathbf{R}^{(t)} \leftarrow \mathbf{M} - \mathbf{X}\hat{\boldsymbol{\beta}}^{(t)}$;
    $\mathbf{r}_k^{(t)} \leftarrow \|\mathbf{R}_k^{(t)}\|_2, \forall k \in [1, K]$;
    $\mathbf{w}_k^{(t)} \leftarrow (1 - \mathbf{o}_k) \odot w_\alpha\left(\frac{\mathbf{r}_k^{(t)}}{\sigma}\right), \forall k \in [1, K]$;
    $\mathbf{W}^{(t)} \leftarrow \underset{k}{\mathrm{diag}}(\mathbf{w}_k^{(t)})$;
    $\hat{\boldsymbol{\beta}}^{(t+1)} \leftarrow \hat{\boldsymbol{\beta}}_{WLS} = \left(\mathbf{X}^\mathsf{T}\mathbf{W}^{(t)}\mathbf{X}\right)^{-1}\mathbf{X}^\mathsf{T}\mathbf{W}^{(t)}\mathbf{M}$;
    $evol = \max_{j\in[1,nm]}\left(\|\hat{\boldsymbol{\beta}}_j^{(t+1)} - \hat{\boldsymbol{\beta}}_j^{(t)}\|_2\right)$;
    $t \leftarrow t + 1; \quad i \leftarrow i + 1$;
  until $(i > nitermax)$ *or* $(evol < evolmin)$;
end
$\hat{\boldsymbol{\beta}} \leftarrow \hat{\boldsymbol{\beta}}^{(t)}$;

Algorithm 1: Robust estimation of $\hat{\boldsymbol{\beta}}$ with IRLS

The scale parameter σ can be estimated alternately with the estimation of $\hat{\boldsymbol{\beta}}$, as in (Yang et al., 2019). However, this solution is known to be rather unstable. Moreover, in our case, the data are quite similar between tunnels. So it makes sense to leave the choice to the user's experience. A value about 1 mm is convenient in most situations. Another important issue is the number of nodes in the spline model. It should be chosen in such a way as to avoid any over- or under-fitting effect. In (Yang et al., 2019), where the task was to fit 1D profiles using evenly spaced nodes, this choice was automated by monitoring a goodness-of-fit criterion. However, the number of nodes is not the only element to be fixed: their positioning is also very important, and can hardly be automated. Here again, we rely on interactivity.

To manage all these aspects, we have developed a user interface, shown in Fig. 4. It is made of five panels. At the bottom, there is a global representation of the tunnel, on which the region corresponding to the data displayed in 3D in the bottom-left panel is indicated. This area represents a surface of about 2×6 m. The top left panel gives access to the parameters of the process. The result is shown as a brown overlay in the bottom-left panel. The top right panel displays either the original intensity or depth images, the map of residuals, or the final IRLS weights. Finally, the bottom right panel allows to visualize a profile along a line represented in green in the top right panel.

In practice, the surface estimation is done in two steps. A coarse estimation pass is first performed considering 1 point out of 30, with alpha between 1 and 0 and 8 nodes in each (u, v) direction (which is the minimum number of nodes for cubic B-splines). A second pass allows to refine the result of the first one, by taking 1 point out of 5, alpha between 0 and -0.7, and more nodes, placed by the user. By carefully positioning the nodes (duplicate nodes can be used to model discontinuities), the surfaces can be adjusted very finely. In the example shown in Fig. 4, a

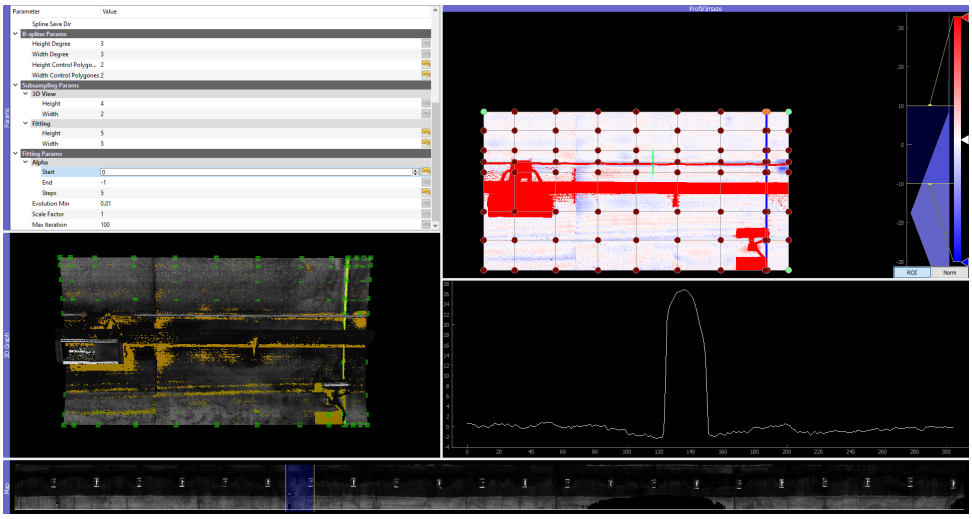

Figure 4. Screenshot of the interface for robust fitting (better visualized by zooming on the electronic version of the figure)

cable with a diameter of about 25 mm can be seen. In an example shown in (Tual et al., 2021), a plate with a thickness of about 5 mm was distinguishable.

3. UNSUPERVISED EQUIPMENT SEGMENTATION

At the end of the robust fitting process described in the previous section, one obtains a map of residuals, defined as the distance between the observations and their prediction by the fitted model. As can be seen on Fig. 5 (left), points on the surface of the wall have small residuals, while objects of interest have centimetric to decimetric residuals. Equipment, mostly located in front of the tunnel surface, has positive values, while hollow objects, such as joints or cavities, have negative values. Invalid/out-of-range data have large, positive residual values. In a first approach to equipment segmentation, we use the apparent homogeneity of residuals on objects (as well as their spatial consistency) to implement an unsupervised clustering technique.

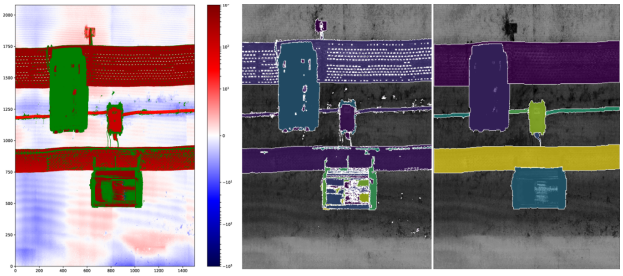

Figure 5. (Left) Residuals of robust surface adjustment. Invalid/out-of-range points appear in green. (Center) segmentation result. (Right) result after manual post-processing using S3A. Each object class appears in a different colour

In (Tual et al., 2021), we proposed to use a Gaussian Mixture Model distribution, whose parameters were estimated by an EM algorithm. A defect of this kind of algorithm is that it is necessary to specify, or estimate, the number of clusters present in the analyzed image. Moreover, the statistics of the classes do not really meet the normality assumption. This is why we have chosen to use the Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) algorithm, proposed in (Campello et al., 2015), that does not require the knowledge of the number of clusters. It is a hierarchical algorithm, of Single-Linkage type, and thus, able to capture clusters of arbitrary shape and sizes. It is also known to be relatively insensitive to noise and outliers. The large size of the processed images, their sometimes cluttered nature, as well as the presence of IOoR data, however, require some adjustments and we have therefore defined a 5-step process.

1. **HDBSCAN clustering on a coarse grid**
   In order to manage the computational issues related to the large size of the images, we first consider subsampled data (1 point out of 6). The feature vector used mixes residual and position information: we consider the triplets $(x, y, \gamma r)$, where γ is a parameter allowing to adjust the relative importance of spatial proximity and residual value coherence. Applying the HDBSCAN algorithm, we obtain a label map, at a coarse resolution.

2. **Label propagation using k-nn**
   In order to recover the original resolution, we apply a propagation technique based on the k-nearest neighbor algorithm. The unlabeled pixels in the full-resolution grid receive the most represented label among those of their k nearest neighbors in the coarse label map. We consider Euclidean distances, computed with the same feature vector as in step 1. At the end of this step, all points of the image (at the original resolution) are in a cluster.

3. **Grouping of background clusters**
   At the end of step 2, the elements of the tunnel lining can be found in several different clusters, which must be grouped together. To do this, we calculate the statistics (mean and variance) of the residual values on each cluster. Clusters with a mean residual of less than 2 mm (in absolute value) and a standard deviation of less than 40 mm are considered as background. If the image contains only the background label, the process is over and the next image can be processed. If not, there are objects of interest and it is necessary to refine the labeling.

4. **Handling invalid/out-of-range (IOoR) pixels**
   IOoR pixels can either correspond to entire, separate objects (see e.g. the light fixture on the top-left of Fig. 5) or belong to existing objects. Therefore, groups of IOoR pixels with an area greater than 200 pixels are given their

own label, while smaller ones or isolated IOoR pixels are assigned to existing clusters. To this end, we first compute an extended label map, by a region growing technique in the image domain (implemented in Scikit's *expand_label* function), considering non-background and non-IOoR clusters only: these are expanded until they touch neighboring clusters or until the maximum enlargement distance is greater than a threshold of 70 pixels. Then, the remaining IOoR pixels receive the (spatially) closest label in the extended label map.

5. **Elimination of small clusters**
   Finally, a morphological opening is performed on the label map and clusters of too small size (area less than 100 pixels) are deleted.

The segmentation process includes a number of parameters (those listed above and those specific to HDBSCAN). These parameters were optimized using quality measures specific to unsupervised classification, such as the adjusted Rand index (ARI), see (Hubert and Arabie, 1985) or the V-measure (Hirschberg and Rosenberg, 2007). The former measures the similarity of two assignments by assessing the classification of pairs of samples, and normalizing for what would have happened by chance. The latter is the harmonic mean of homogeneity (each cluster should only contain members of a single class) and completeness (all members of a given class should belong to the same cluster). Although optimized, the values obtained in our experiments remain rather low (typically, about 0.61 to 0.65 for both metrics, that range in [0,1]). This confirms the tendency to over-segmentation that can be observed in the image of Fig. 5 (center).

The segmentations can be manually edited using specialized software such as S3A (Jessurun et al., 2020). A manual intervention is anyway necessary to assign a semantic label (e.g. wire, light, safety sign) to each identified cluster. An example of corrected result is shown in Fig. 5 (right). It is therefore not possible to completely avoid user interaction and this makes the process impractical for industrial routine. However, the method is more efficient than a purely manual segmentation. We used it to annotate data to develop the supervised learning method that will be described in the next section.

4. SUPERVISED EQUIPMENT SEGMENTATION

The supervised deep learning architecture Mask R-CNN (He et al., 2017) appears well adapted for image-based equipment segmentation. Indeed, this approach allows to distinguish multiple instances of objects, with possible overlaps. To obtain a segmentation of object instances, the algorithm first extracts feature maps from an input image, by a succession of convolution, pooling. It uses dropout operations to avoid over-fitting. In a second step, bounding boxes and classes of objects of interest are extracted according to a process derived from the Faster R-CNN architecture (Ren et al., 2015). Compared to the latter, Mask R-CNN also computes the pixel-by-pixel mask of each object of interest in a bounding box. This second step operates at several scales to perform the segmentation of objects of different sizes. In this paper, we implement the Mask R-CNN architecture using only intensity images to perform a multi-class segmentation. In table 1, we list the 14 equipment categories that have been identified. Note that some elongated objects, such as cables or cable trays, will appear in pieces on a series of contiguous images. Others have a smaller extension (lighting, signs...) and will be visible, at best on a single image, but sometimes in two pieces on two successive images.

cable	503	joints	132
lightings	84	cable tray	121
SOS exit	19	electric box	85
SOS reflective sign	12	traffic sign	5
front emergency sign	2	SOS ID	10
emergency exit sign	19	formwork	3
emergency exit	4	location plate	4

Table 1. Equipment list and number of equipment occurence.

For training and evaluating the Mask R-CNN model, we built a database of 415 images. We divide the database according to the usual train-validation-test paradigm as follows: 64% of the images for learning, 20% for validation and best model selection, 16% for testing. It may be noticed that the different classes are very imbalanced. The number of object occurrences can range from a few units for some categories to several hundred for others. As is usual for deep learning networks, we use a pre-trained model to extract the feature maps. The learning process is therefore focused on the instance segmentation module from the training images. In the model learning phase, data augmentation methods are applied to increase the variability of the data set and thus improve model performance. For this purpose, at each epoch we apply a transformation (Gaussian noise, median blur, contrast variations, rotation, image compression) on each image. The values applied are chosen randomly within a pre-defined range. Among the 150 training epochs, we select the model that minimizes the loss function of Mask R-CNN.

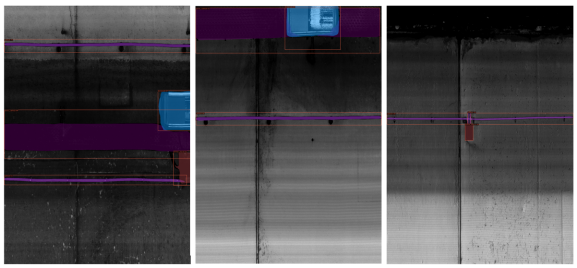

Figure 6. Instance segmentation result without any postprocessing operations

The first results, based on this rather limited database, show that segmentation performance is class-dependent. On the examples in Fig. 6, we can observe that the wires, cable trays, lighting and electric box are well identified, but the model does not detect the joint visible in the middle of the images. A quantitative evaluation confirms that objects such as cable trays, lights, or decametric plates, which are numerous in the database, are relatively well detected: we observe a good true detection rate, around 95% for each class, with a precision around 80%. The results obtained on less represented objects are very variable, and it is necessary to increase the corpus of data to be able to draw conclusions. Finally, joints are, surprisingly, little detected. This effect is under investigation, but we think that the use of depth in addition to the image information could be interesting for this type of structural element.

Without any post-processing operations, we observe that many objects appearing only partially in the images are not correctly detected and identified. To overcome this issue, we supplement

the test datasets by generating "in-between" images with 50% overlap, as shown in Fig. 7. During the inference stage, the selected model is applied to this augmented test data set. Objects must be detected in at least two contiguous images to be retained. Two situations are possible :

- When an object is detected at most three times in contiguous images, we consider it to be a small-size object. In this case, we keep the predicted object whose bounding box center is the most central in the image. For example, in Fig. 7, we can see that the lighting is detected, at least partially, in the three images. The bounding box of lighting in the bottom image is the most central. In the end, only this one is retained.
- When an object is predicted in more than three images and the bounding boxes of the detected objects reach the left and right edges of each image, it is most likely a linear object, such as a cable. In this case, the predictions are joined to form a single object in an assembly of contiguous images, as illustrated in Fig. 8.

Thanks to these post-processing operations, the performance seems visually better. However, this first impression needs to be consolidated by a quantitative evaluation, which is the subject of an ongoing work.

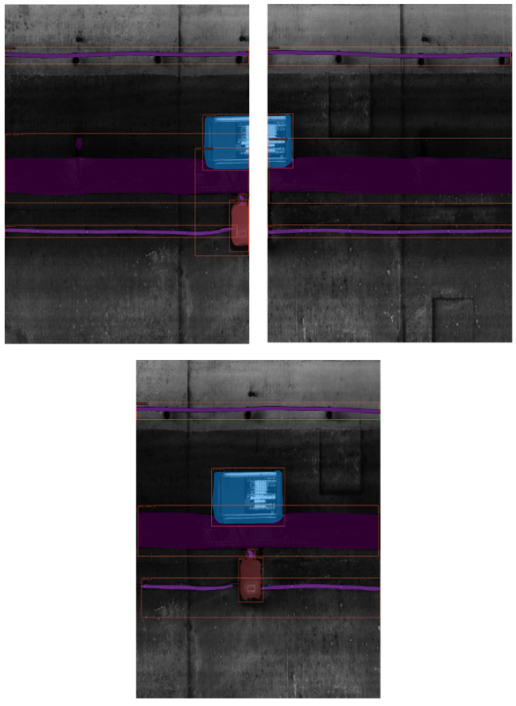

Figure 7. Results obtained on two contiguous images (top) and on their "in-between" image (bottom)

5. DISCUSSION AND CONCLUSIONS

In this contribution, we have introduced an approach that allows the extraction of semantic information on a metric (tunnel shape) and centimetric (equipment) scale, based on the analysis of depth maps acquired in tunnels. We also explored a deep learning approach for tunnel equipment segmentation from images.

First, we described a method for estimating the control points of B-Splines surfaces using a parametric family of M-estimators, which achieves a high degree of robustness to outliers caused by invalid/out-of-range data, artifacts, and equipment. The resulting tunnel wall modeling can be made accurate enough to distinguish objects of interest a few millimeters thick, provided the node points of the B-Spline are judiciously placed. For now, we rely on user interaction to position these points. It is necessary to place enough points to have a good approximation of the surface, while avoiding over-fitting; it is necessary to duplicate some nodes to correctly model surface discontinuities. At the same time, care must be taken not to position node points on the equipment to avoid modeling them as if they belonged to the surface. All this can lead to dilemmas and makes the automation of node placement a difficult *model selection* task. To go beyond the method proposed in (Yang et al., 2019) in a relatively simple setting, that relies on traditional figure-of-merit monitoring techniques, the use of machine learning is an appealing research issue (Laube et al., 2018). We aim to investigate techniques that use image data to train an algorithm on where to position control points in order to fit a spline surface on depth data. Another, more challenging research prospect is to develop deep learning models capable of fitting the tunnel shape directly from the image and depth data, without human intervention and in a robust manner.

Modeling the surface of tunnels can be useful in itself for structural geometry control applications. It also enables the correction of raw depth information, making it easier to distinguish small objects. Unfortunately, unsupervised segmentation based on residual depth information alone leads to over-segmentation of objects of interest, as we have experienced. This processing step therefore also requires user intervention.

In this paper, we also experimented on a supervised instance segmentation deep learning model, namely Mask R-CNN based on image data. The first results, obtained from a rather limited database, are encouraging but show that the segmentation performances depend on the classes, and in particular on their number in the training set. These results must be rapidly consolidated, based on an increased corpus of data. In addition, the post-processing still needs to be finalized so that the method can be evaluated at the scale of an entire structure, and not only on individual images.

Finally, the segmentation techniques we have deployed are based on either depth data or image data, and we see that neither of them gives complete satisfaction. The joint use of 3D and image is a good direction of improvement in supervised instance segmentation, but raises questions about the best way to combine the two modalities. The most obvious approach, called *early fusion*, is to combine the two into a two-channel image. It raises the question of the normalization of data, since depth and intensity have very different ranges. Our approach of correcting the depth data for the overall tunnel shape, that requires significant human intervention might not be fully justified in this context. We might imagine using less precise information, but requiring a lesser degree of supervision. Finally, other fusion techniques, like *late fusion*, as proposed in (Aakerberg et al., 2017), would probably be less sensitive to data normalization. All these aspects will be investigated in the near future.

ACKNOWLEDGEMENTS

The authors would like to thanks Pavemetrics for their support in data acquisitions.

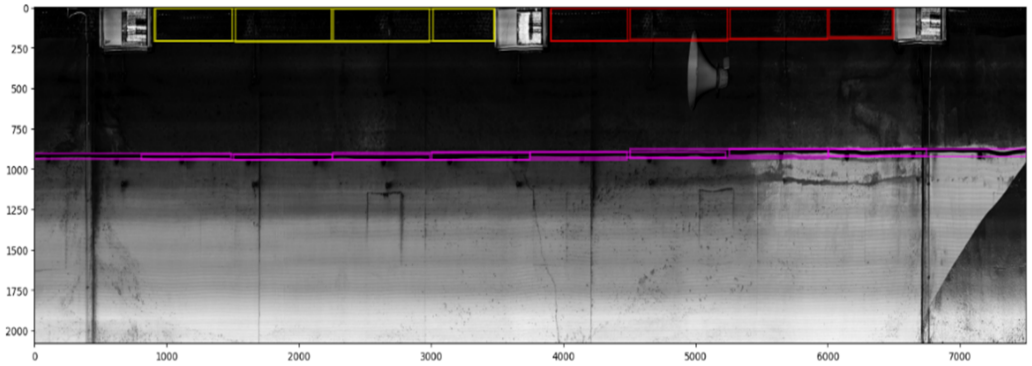

Figure 8. Merging of linear object pieces detected in contiguous images.

REFERENCES

Aakerberg, A., Nasrollahi, K., Rasmussen, C. B., Moeslund, T. B., 2017. Depth value pre-processing for accurate transfer learning based RGB-D object recognition. *Procs. 9th International Joint Conference on Computational Intelligence*, Funchal, Madeira, Portugal, 121–128.

Campello, R. J. G. B., Moulavi, D., Zimek, A., Sander, J., 2015. Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection. *ACM Trans. Knowl. Discov. Data*, 10(1). Article n°5, 51 pp.

Foucher, P., Charbonnier, P., Noël, T., Fosse, Y., J.-F.Hébert, 2019. Scanning tunnels with two very high-resolution laser devices and a stacker. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, XLII-2/W18, ISPRS Workshop "Optical 3D Metrology (O3DM)", Strasbourg, France, 39–46.

Geman, S., McClure, D., 1985. Bayesian image analysis: an application to single photon emission tomography. *Proc. Statist. Comput. Sect., Amer. Statist. Assoc.*, Washington, DC, USA, 12–18.

Gui, R., Xu, X., Zhang, D., Pu, F., 2019. Object-Based Crack Detection and Attribute Extraction From Laser-Scanning 3D Profile Data. *IEEE Access*, 7, 172728-172743.

Gupta, P., Dixit, M., 2022. Image-based crack detection approaches: a comprehensive survey. *Multimedia Tools and Applications*, 81, 40181—40229.

Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., Stahel, W. A., 1986. *Robust Statistics - The Approach Based on Influence Functions*. Wiley.

He, K., Gkioxari, G., Dollár, P., Girshick, R., 2017. Mask R-CNN. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2980–2988.

Hirschberg, J., Rosenberg, A., 2007. V-measure: A conditional entropy-based external cluster evaluation. *Proc. Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 410–420.

Holland, P., Welsch, R., 1977. Robust Regression using Iteratively Reweigthed Least Squares. *Communication in Statistics – Theory and Methods*, A6(9), 813–827.

Hubert, L., Arabie, P., 1985. Comparing partitions. *Journal of Classification*, 2, 193–218.

Jessurun, N., Paradis, O., Roberts, A., Asadizanjani, N., 2020. Component Detection and Evaluation Framework (CDEF): A Semantic Annotation Tool. *Microscopy and Microanalysis*, 26(S2), 1470–1474.

Lam, S. K., Pitrou, A., Seibert, S., 2015. Numba: A LLVM-based python JIT compiler. *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, LLVM '15, Austin, Texas.

Laube, P., Franz, M. O., Umlauf, G., 2018. Deep learning parametrization for B-Spline curve approximation. *Procs. International Conference on 3D Vision (3DV)*, 691–699.

Laurent, J., Hébert, J., 2002. High performance 3D sensors for the characterization of road surface defects. *Proceedings of the IAPR Workshop on Machine Vision Applications*, Nara, Japan.

Nguyen, S., Tran, T., Tran, V., Lee, H., Piran, M. J., Le, V. P., 2022. Deep Learning-Based Crack Detection: A Survey. *Int. Journal of Pavement Research and Technology*. In press.

Piegl, L., Tiller, W., 2012. *The NURBS book*. Springer, Berlin / Heidelberg.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. *Procs. Advances in neural information processing systems*, 1, 91–98.

Tarel, J., Ieng, S., Charbonnier, P., 2002. Using robust estimation algorithms for tracking explicit curves. Springer (ed.), *6th European Conference on Computer Vision (ECCV), Lecture Notes in Computer Science*, 2350, Copenhagen, Denmark, 492–407.

Tual, M., Charbonnier, P., Foucher, P., 2021. Robust B-spline surface estimation for tunnel lining modelling and equipment surveying. *Near Surface Geoscience Conference & Exhibition 2021 (NSG'21)*, EAGE, Bordeaux, France.

Xu, X., Yang, H., 2020. Vision Measurement of Tunnel Structures with Robust Modelling and Deep Learning Algorithms. *Sensors*, 20(17). Article n°4945.

Xu, Y., Li, D., Xie, Q., Wub, Q., Wang, J., 2021. Automatic defect detection and segmentation of tunnel surface using modified Mask R-CNN. *Measurement*, 178. Article n°109316.

Yang, H., Xu, X., Kargoll, B., Neumann, I., 2019. An automatic and intelligent optimal surface modeling method for composite tunnel structures. *Composite Structures*, 208, 702-710.