

Active inspection of objects to detect possible damage and measure their volume using 3D reconstruction

Vladimir Riffo^{a,*}, Rodrigo Hidalgo^a

^aDepartamento de Ingeniería Informática y Ciencias de la Computación, Universidad de Atacama, Copiapó, Chile.

ARTICLE INFO

Keywords:

Volume measurement
Active inspection
Damage detection
3D Reconstruction
Non-Destructive testing

ABSTRACT

Automated inspection processes offer the advantages of objectivity and efficiency for each test. For this reason, certain techniques for digital image processing and machine vision have been developed. However, it remains an open question due to the great variability in appearance and shape of the objects of study. In this work, we propose an inspection methodology that allows us to analyze objects for which we have developed an active inspection system to detect damage or malfunction. Its operation begins with the acquisition of images of the object using two cameras located in front of a rotating platform, with a controlled rotation, and projecting a laser line on the object. The following is the 3D reconstruction of a section of the object, to then make a comparison using a 3D descriptor and distance analysis between points of the 3D test model and the 3D model generated with a base version of the same object without damage, which was stored in a database. If any damage is detected, the inspection is continued until the entire damaged section is completed and its volume is calculated. The main contributions of our work are summarized as: *i*) Efficient combination of machine vision algorithms for volume estimation, *ii*) Active inspection in the automatic search for surface damage on objects, and *iii*) Measurement of the damage volume detected in metric units, which can be subtractive or additive damage. To validate our proposal, we have applied this methodology to different objects, thus being able to estimate the volume of damage (if any). Our results are promising, as we achieved a Mean Absolute Percentage Error, $MAPE = 2.78\%$.

1. Introduction

“Eureka”, would have been the word used by the Greek mathematician Archimedes of Syracuse after discovering what is now called Archimedes’ Principle, which establishes the relationship between the volume of a submerged body and the buoyant force that it experiences. The above demonstrates the importance and complexity of determining the volume of an object and thus verifying its integrity.

Inspection of objects as a human task is vulnerable to errors and, as a result, automation of inspection systems becomes essential to improve the effectiveness of results [1]. The literature shows that different approaches have been generated in the area of automatic object inspection, depending on the application [2]. However, automatic visual inspection still has issues that need to be addressed: *i*) *loss of generality*, this is because approaches developed for one application cannot be used in others; *ii*) *poor detection accuracy*, which means that there is a fundamental trade-off between false positives (false alarms) and loss of detections; *iii*) *limited robustness*, since the requirements for using a method are often met only for simple structures; and *iv*) *low adaptability*, as it can be very difficult to modify the design of an automatic system [3].


In recent years, visual inspection and damage detection methods have improved in the manufacturing industry with systems that focus on the use of artificial neural networks [4, 5] or those that use stereoscopic vision and scanning with

laser line projection in different types of scenes, along with a design that allows greater mobility of the object to be analyzed, in order to achieve a more complete 3D reconstruction [6, 7, 8].


However, these last types of systems generally have some disadvantages such as erratic operation, a large amount of time, limited use and the need for trained and experienced workers to operate them; *e.g.*, in applications that allow inspection of complex pipelines, where a fast and accurate 3D reconstruction solution is required, for the inspection of the external shape of the pipes before assembly [9].

In this work, we propose a methodology that adequately merges different computer vision and active search techniques to perform an automatic visual inspection, which does not suffer from the four problems mentioned above. We believe that our methodology is a useful and powerful alternative to examine different types of complex objects in a general, precise, robust and adaptable way because we use a novel strategy: using a rotating platform to place the test object, we begin with the acquisition of a number of relevant initial views, where each view contains a laser line beam, deformed according to the topology of the object using two cameras (stereoscopic vision), to then perform the 3D reconstruction, and later, compare this reconstruction with an ideal model. If something suspicious was detected, we will investigate further to determine the full area of possible damage that the test object could have and calculate its volume. If no damage is found in the initial views, we continue with the inspection until the entire test object is visually covered. Applying this proposal, we have been able to estimate the volume of two types of damage: *a*) *subtractive damage*, which consists of volume loss in the test object; and *b*) *additive damage*,

*Corresponding author

 vladimir.riffo.b@gmail.com (V. Riffo);

rodrigo.hidalgo@alumnos.uda.cl (R. Hidalgo)

 <http://www.diicc.uda.cl/profesores/vriffo/index.html> (V. Riffo)

ORCID(s): 0000-0003-1274-6224 (V. Riffo)

which consists of the incorporation of unwanted volume to the test object. Our results lead us to think that this proposal is: *i)* an innovative method for automatic visual inspection of objects; *ii)* an efficient visual inspection method; *iii)* an adequate fusion of computer vision and active search algorithms, applied to inspection problems; and *iv)* a robust and generic methodology for the inspection of complex objects.

In summary, since we have efficiently combined different machine vision algorithms to perform an active inspection of objects in a non-destructive way, in this work we propose an innovative method; the test object is rotated until a damage is found -if any-, after that, the object continues rotating until the damage is completely described. Once the damaged area of the object has been fully segmented, it is reconstructed in 3D and the damage volume is calculated by comparing and matching it with a damage-free 3D reconstruction of the same object. With our proposal, we can inspect objects without damage, with additive damage or with subtractive damage.

2. Related Works

Advances in the area of non-destructive inspection using computer vision show a growing potential due to the different approaches that have been generated over the years [2], in particular, the development of systems that can be robust, accurate and efficient when inspecting objects.

A real-time quality inspection system for metal products and flat metal sheets was developed by [10]. This system is based on optical triangulation, using a laser line pattern, which provides the position of the light projected on the surface of the objective, through a simple geometry. The steel strips move through a pass band, where a camera with a CMOS sensor (Complementary Metal-Oxide-Semiconductor) captures the projection of a laser line, in order to reconstruct the surface in three dimensions and thus analyze the two important geometric factors in the metallic strips: the flatness and the width. Three years later, in 2013, [11] implemented improvements in detection and extraction performance, incorporating new types of processing: one of high speed but less precise and another that provides high precision while increasing times calculation. A mechanism was also proposed to reduce effects on the movement of metal sheets during the manufacturing process.

An interesting approach implemented an automated inspection system that uses a rescan strategy, which allows to minimize operating time [12]. By inputting a CAD model of the object to be inspected, the system begins by scanning the entire area of the object and develops an approximate model to calculate a scan path. Then, the complete scan phase of the test object is carried out to reconstruct it and thus generate an error map, which indicates the areas that are represented with large errors or that were not scanned. In its last phase, the areas indicated as deficient by the error map are re-scanned to generate an improved error map and then inspect the object with greater reliability. In the same year, an inspection system was implemented using stereo vision

and laser triangulation for reconstruction [13]. One of its great advantages lies in not being dependent on the amount of light in the environment present at the time of inspection. Its implementation is simple: two stereo cameras and a laser emitter that hit the object in order to capture its silhouette and then obtain enough information to reconstruct. A year later, they optimized said system so that it was capable of detecting the presence of damage to the inspected object [14], showing great precision in the results obtained.

Recently, a methodology for the detection of damaged fruits using millimeter wave (mmwave) measurement [15] was implemented. It detects the relative permittivity difference included in the millimeter wave images between healthy and damaged fruit. Subsequently, the support vector machine (SVM) algorithm is used, which is a Machine Learning Algorithm (MLA) method able to classify healthy fruits from damaged fruits. The system that was implemented to validate the methodology consists of a spherical 3D scanner, a Rohacell tower where the fruit is placed, and a probe antenna that rotates on the fruit with a radius of 585 mm. To minimize the measurement time, which reached 2 hours, and the amount of information (number of measurement points) used, the Gray Wolf Optimizer (GWO) method was implemented in order to find optimal values necessary for learning the SVM algorithm. In their research results, they obtained a precision between 93% and 100% in the tests carried out with peaches and apples, both damaged and healthy. This work shows preliminary and inconclusive results as it is a static inspection model for the detection of damage in fruits and it does not show a method for the quantification of the detected damage. It is a different work from ours as we use different algorithms and means to obtain information.

A similar proposal in some respects has been shown in [16], *i.e.*, a surface inspection methodology using computer vision techniques. In this work, a system consisting of high-precision cameras integrated with laser scanners to capture surface data was designed. Subsequently, with the information captured from the object, an object detection method is executed by means of the voxelization of the point cloud obtained from the laser scanning and then, skeletonization algorithms are used for the created voxel, obtaining a representation of the scanned object. Afterwards, it is compared with a model already defined in a database. That comparison analysis is based on the alignment of both skeletons, that is, the one of the analyzed object and the one belonging to the database. Once analyzed, and if any damage is found, the voxelization already created is used to extract the dimensional value of the detected damage. The main difference with our method is that it is a static method, designed for the detection and quantification of damage to large surfaces.

An interesting proposal for the agricultural industry [17], shows a computer vision system for the estimation of apple volume and weight by using 3D reconstruction and non-contact measuring methods. The 3D surface of the apples was reconstructed using a single multispectral camera and a linear-array structured light. Both the traditional image feature and height information were extracted from the height

maps. Two different types of height features were extracted, and both of them were fused with a projection area to form combination features. The system and method developed in this study provide an alternative to the traditional methods for noncontact measurement of the volume and weight of agricultural products. This work performs the 3D reconstruction in a similar way to our proposal and instead of using a rotating platform, it uses a convey belt. However, it is not focused on automatic damage detection and quantification.

An application to measure the surface damage volume of structures is shown in [18]. This study proposes a method based on 3D structural surface damage reconstruction techniques to reconstruct and extract data for damage volume calculation. The surface damage of concrete specimen is three-dimensionally reconstructed using multi-view images taken with smartphones and compared with a depth camera. The point cloud data was obtained, and then the damage plane was fitted and removed by a Random Sample Consensus algorithm to obtain the damage site data; finally, the damage volume was calculated. To determine the volume of damage, the difference between the volume reconstructed with the flat surface without damage is made in a similar way to our proposal. In this proposal, neither active inspection nor automatic detection of the damaged area is considered, and consequently discriminate whether there is or not a damage. In addition, our method considers the inspection of objects with irregular shapes and we make a complete inspection of said objects.

After reviewing the literature and related works, we can say that 3D reconstruction is widely developed and different strategies are used to carry it out. The two main strategies are the use of multiple views and structured light projection. Additionally, we have seen that damage detection is still an open question, with a wide variety of algorithms and strategies ranging from image processing to the use of convolutional neural networks (CNN). Due to the difficulties involved in calculating volume, few works focus on this problem. We have not found proposals in the literature that efficiently combine strategies that allow performing an automated visual inspection of objects to actively look for any damage in them (as a human would do), and that incorporate the quantification of the damage. We think that our proposal is innovative, generic, and easily adaptable and it provides auspicious results.

3. Proposed Method

The methodology used in this proposal is based on a modification of Knowledge Discovery in Databases (KDD). This is a well-known five-step methodology; Data acquisition and selection, Pre-processing, Processing, Classification and Evaluation [19].

The methodology developed for the inspection of objects using computer vision and active search is shown in algorithm 1. In this algorithm, the parameters *lim1* and *lim2* are the boundaries of *Dmg_Obj*. These values are calculated by first finding the boundary point(s), both left and right,

and then, determining in which step they were obtained during the image acquisition stage. These parameters, *lim1* and *lim2*, are used to determine whether *Dmg_Obj* is incomplete and in which direction of rotation the new iteration of the object analysis will be performed, in order to complete the damaged area. This can be defined as follows: if *lim1* is equal to *IStep* it means that the damaged area is incomplete in the left sector and therefore, a new analysis must be performed, rotating counterclockwise. On the other hand, if *lim2* is equal to *FStep* it means that the damaged area is incomplete on the right side and the object must be analyzed clockwise. If *lim1* is different from *IStep* and in turn, *lim2* is different from *FStep*, it means that the limits of *Dmg_Obj* are not close to the steps performed during the analysis and the damage is completely defined.

It is then an active inspection process, which starts with the test object arranged in different initial positions, and then actively searches for damage, causing clockwise or counterclockwise rotations as described above, until the damage is completely delimited and can be calculated (see results of this work in Table 1).

3.1. Inspection Method

Choosing the appropriate inspection method is essential for the development of our methodology since from this choice we will determine the three-dimensional reconstruction mechanism of the object of interest.

In general, there are two sets of inspection: *i*) those of direct contact with the object by means of a probe that runs along its surface, transferring the coordinates of each point to the computer. This probe displacement system gives a high resolution to the reconstructed model but it turns out to be very slow and could eventually be detrimental to the object under study because contact with the probe can damage its surface [20]; *ii*) eliminating the need for direct contact with the object, focusing its interaction on other aspects, *e.g.*, optical interaction. These are called Non-Destructive Tests (NDT) and are used in different areas because they can be performed during any of the life stages of the object [21]. This last set is more focused on the evaluation and detection of superficial damages and therefore it is the most suitable option.

Within this group, the visual method is the most suitable due to its simplicity when implementing and manipulating.

3.2. Image Acquisition

With the inspection method chosen, we need to delve into the image acquisition technique that allows us to transform an image into a data set that can be manipulated [22]. For our case, these data will help us to generate a three-dimensional reconstruction of the object.

The active triangulation technique makes it possible to determine the 3D topography of the object of interest using a geometric principle: rays of light, which can be emitted by a digital video projector, an analog slide projector or a laser [23] that falls on the surface of the object of study [24]. We have decided to use this technique as, for an automatic visual

Algorithm 1: Inspection methodology

Input: $IStep = 1$ %Initial step, $FStep = 50$
%Final step, $dir = 1$ %Rotation direction,
 $ncaps = 50$ %number of captures made, P
%Projection matrix, $TObj$ % 3D
Reconstruction of the target object.

Output: Vol %Damage volume.

```
1 while  $ncaps < 200$  do
2   %Stage 1: Image Acquisition
3    $[LOn, LOff] \leftarrow$ 
    Acquire_Img( $IStep, FStep, dir$ );
4   %Stage 2: Pre-processing
5    $[Img\_Skel, ] \leftarrow$  Pre-proc( $LOn, LOff$ );
6   %Stage 3: 3D Reconstruction
7    $[TObj, RGB\_TObj] \leftarrow$ 
    Reconstruction_3D( $dir, Img\_Skel, P$ );
8   %Stage 4: 3D PointCloud Alignment
9    $Align\_SObj \leftarrow$  Alignment( $TObj, SObj$ );
10  %Stage 5: Damage Detection
11   $Dmg\_Obj \leftarrow$  Detection( $Align\_SObj, TObj$ );
12  if  $Dmg\_Obj \neq 0$  then
13    if  $lim1 = IStep$  then
14       $IStep = FStep + 1$ ;
15       $FStep = FStep + 15$ ;
16    else
17      if  $lim2 = FStep$  then
18         $IStep = FStep + 1$ ;
19         $FStep = FStep + 15$ ;
20         $dir = -1$ ;
21      else
22        %Stage 6: Volume Calculation
23         $Vol \leftarrow$  Calc_Vol( $Dmg\_Obj$ );
24      end
25    end
26     $ncaps = ncaps + 15$ ;
27  else
28     $IStep = FStep + 1$ ;
29     $FStep = FStep + 30$ ;
30  end
31 end
```

inspection system, it is relatively simple to implement [25]. Besides, it has been decided to incorporate stereo vision.

The proposed system for an adequate acquisition of images of the object of interest requires: a rotating platform, composed of a stepper motor that allows movement control; two cameras in stereo configuration, with the possibility of being able to configure their position, in order to modify the point of view; one or two laser line emitters and a computer, which will be in charge of carrying out the image acquisition and storage process (see Figure 1).

To perform the 3D reconstruction of the objects arranged on the rotating platform, it is necessary to calibrate the cameras prior to the inspection process. This will allow obtaining the intrinsic and extrinsic parameters of the cameras.

The intrinsic parameters indicate the coordinates of a 2D point (pixel) in the image from the position of the 3D point in the scene and are; the focal length, the optical center and the distortion coefficients [26]. The extrinsic parameters measure the position and orientation of the camera with respect to the coordinate system established for the world, and are rotation and translation.

Among the existing calibration methods, the [27] method is taken as a reference. This method uses the coordinates of the points located on a calibration pattern, similar to a checkerboard. The images of the calibration standard are acquired considering different positions and orientations.

Once the structure of the image acquisition is defined, the parameters $IStep$ and $FStep$ are defined as the initial and final step of rotating platform, respectively, and dir as the direction of rotation. Image acquisition of the object is performed using the sequence of actions shown in algorithm 2.

Algorithm 2: Function Acquire_Img

Input: $IStep = 1$ % Initial step, $FStep = 50$
%Final step, $dir = 1$ %Rotation direction.

Output: LOn %Vector containing captures of the
object with projected laser line, $LOff$
%Vector containing captures of the object
without projected laser line.

```
1  $[LOn, LOff] \leftarrow$  Acquire_Img( $IStep, FStep, dir$ )
2   for  $n = IStep$  to  $FStep$  do
3     Rotate turntable  $n$  steps in  $dir$  direction;
4     Turn on laser diode;
5      $LOn(n, :) \leftarrow$  Capture image;
6     Turn off laser diode;
7      $LOff(n, :) \leftarrow$  Capture image;
8   end
9 end
```

3.3. Image pre-processing

It is necessary to purify the images obtained before performing the 3D reconstruction to prevent that any unwanted factor present in them, such as differences in lighting, shadows, or distortion generated by the camera lens [28, 29], can generate inconvenience to later stages.

To correct the presence of radial distortion in each image L_On and L_Off (with laser on and off, respectively), caused by the camera lens, we use the radial distortion coefficients obtained from the process of camera calibration. To obtain the undistorted equivalent from each pixel located in the image we solve the coordinates (x, y) from the equations (1) and (2) respectively:

$$\check{x} = x(1 + k_1 \cdot r^2 + k_2 \cdot r^4 + k_3 \cdot r^6), \quad (1)$$

$$\check{y} = y(1 + k_1 \cdot r^2 + k_2 \cdot r^4 + k_3 \cdot r^6), \quad (2)$$

where, (\check{x}, \check{y}) are the distorted coordinates of the pixel and

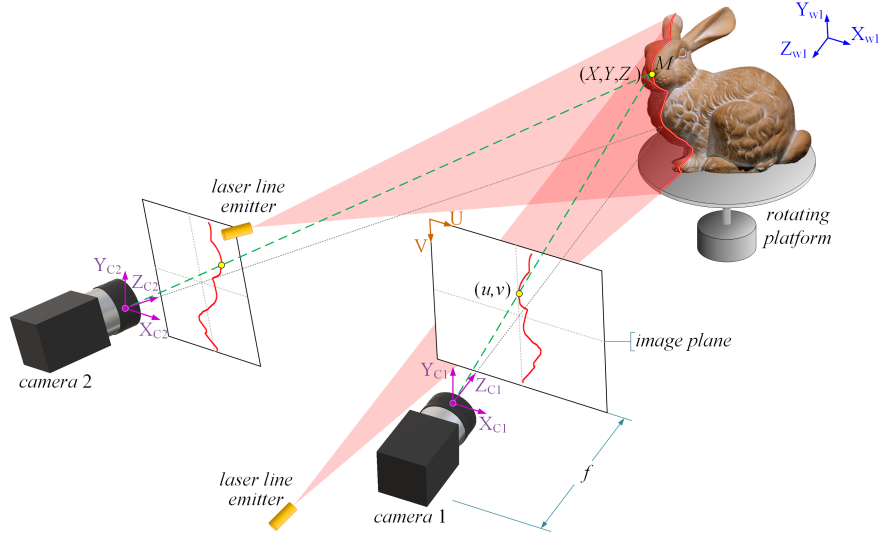


Figure 1: Inspection system to detect possible damage.

(x, y) are the undistorted coordinates. k_1, k_2 and k_3 are the radial distortion coefficients and $r = \sqrt{x^2 + y^2}$.

From the previous action, we get the set of undistorted images ND_LOn and ND_LOff . For each pair of images associated with a given step.

After eliminating the radial distortion produced by the camera lens and obtaining ND_LOn and ND_LOff , it is necessary to segment the laser line, for which the respective red channels (R_LOn and R_LOff) are extracted from ND_LOn and ND_LOff images and then they are subtracted, obtaining as a result an image where only the projected laser line is displayed, named Img_Laser , as shown in Figure 2. Once the laser line has been segmented, it is necessary to obtain the skeleton of said line, for which, each pixel of Img_Laser is analyzed in search of the maximum intensity value present, v_max . With this data we calculate the minimum intensity value: $v_min = 0.895 \cdot v_max$, where the scalar 0.895 was empirically determined. With these values of v_min and v_max , we calculate the appropriate threshold value U to binarize Img_Laser , using $U = v_max - v_min$.

With Img_Laser binarized, we apply an erosion using a disk of 1 mm radius as the object of the structuring element, and thus, skeletonization is finally done. To obtain the skeleton of the laser line projected on the object, we generate a region of interest in the image to determine the location of the line and discard the environment of the scene. The result of the skeletonization process and the previous steps can be seen in Figure 3.

For the damage detection stage, it is important to identify each pixel of the laser line skeleton in the ND_LOff image in order to store the respective *RGB* color set (*Red*, *Green*, *Blue*) of the surface of the object on which the laser line was projected.

All this is summarized in algorithm 3.

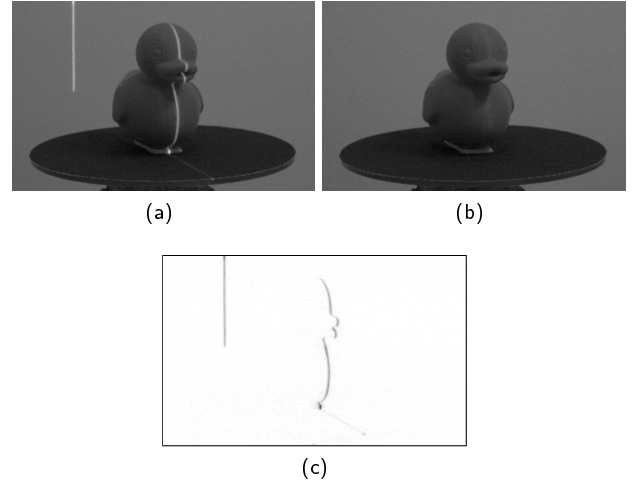


Figure 2: (a) Red channel of ND_LOn , (b) Red channel of ND_LOff , (c) Subtraction of red channels ($R_LOn - R_LOff$).

3.4. 3D Reconstruction

To carry out the 3D reconstruction of the skeleton of the laser line, formed by the points (u, v) , *i.e.*, to carry out the transformation $(u, v) \mapsto (X, Y, Z)$, we will use the geometric model shown in Figure 1. We will only explain the procedure that allows the 3D reconstruction of the images acquired with camera 1, since the procedure for camera 2 is exactly the same.

We will begin by explaining the geometric model, where initially the relationship of a 3D point and its projection in an image as a 2D point is observed (3D mapping \mapsto 2D), which can be expressed as a linear mapping in homogeneous

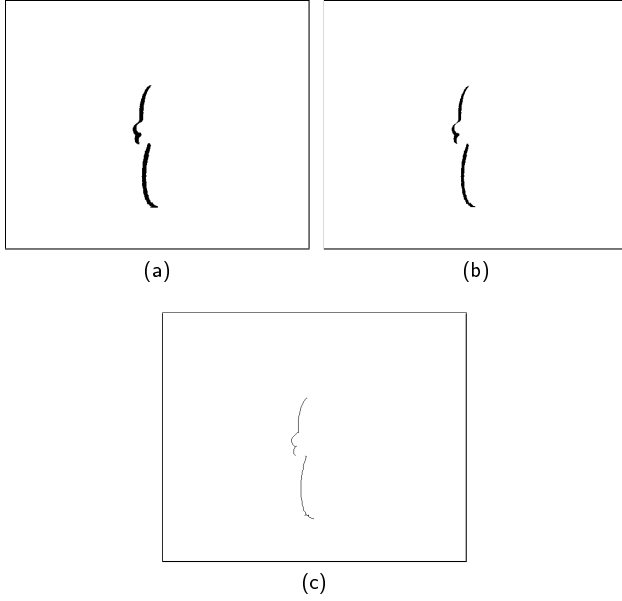


Figure 3: (a) Binarization of *Img_Laser*, (b) Erosion using a disk of 1mm radius, and (c) Skeletonization.

coordinates:

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = P \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (3)$$

where, λ is a scale factor, P is the projection matrix of 3×4 , points (u, v) and (X, Y, Z) are the 2D and 3D coordinates respectively for each point on the laser line. The parameters included in the P matrix can be estimated through a calibration process [27].

Using A and B (of the linear equation system $Ax = B$) we obtain the 3D points (X, Y, Z) of the laser line, with respect to the world coordinate system (X_{w1}, Y_{w1}, Z_{w1}) , as follows:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = A \setminus B \quad (4)$$

Since the inspected object is on a rotating platform, this rotation must be considered for each acquisition and the displacement must be incorporated at the points (X, Y, Z) . Then, incorporating a new transformation $(X, Y, Z) \mapsto (X_p, Y_p, Z_p)$, as follows:

$$\begin{bmatrix} X_p & Y_p & Z_p \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}^T \cdot R_p + t_p \quad (5)$$

where, R_p is the rotation matrix and as the rotating platform

only generates rotational movement, the translation vector is considered as $t_p = 0$. Then, R_p is defined as:

$$R_p = R_{px}(\alpha) \cdot R_{py}(\beta) \cdot R_{pz}(\gamma) \quad (6)$$

where, R_{px} , R_{py} and R_{pz} are the rotation matrixes around each coordinate axis of the rotating platform, given the angles α , β and γ respectively.

Since the rotation is only performed for the Y axis, *i.e.*, $\alpha = 0^\circ$ and $\gamma = 0^\circ$, so the rotation angle β is a variable that must be calculated as:

$$\beta = stp \cdot (1.8^\circ \cdot dir) \quad (7)$$

where, stp indicates the number of the step taken when acquiring the image ($stp = 1, 2, \dots, 200$) and dir is the rotation direction given by the rotating platform, if the rotation

Algorithm 3: Function Pre-proc

Input: *ncaps* = 50 %Number of captures made,
LOn %Vector containing captures of the
object with projected laser line, *LOff*
%Vector containing captures of the object
without projected laser line.

Output: *Img_Skel* %Vector containing the
skeletonized laser lines, *RGB_Skel*
%Vector containing RGB Values of the
skeletonized laser lines.

```

1 [Img_Skel, RGB_Skel] ←
  Pre-proc(LOn, LOff, ncaps)
2   for n = 1 to ncaps do
3     ND_LOn(n, :) ← Remove distortion from
      LOn(n, :);
4     ND_LOff(n, :) ← Remove distortion
      from LOff(n, :);
5     R_LOn(n, :) ← Extract red channel from
      ND_LOn(n, :);
6     R_LOff(n, :) ← Extract red channel from
      ND_LOff(n, :);
7     Img_Laser(n, :) ← R_LOn(n, :
      ) - R_LOff(n, :);
8     v_max ← Calculate maximum intensity
      value in Img_Laser(n, :);
9     v_min ← 0.85 · v_max;
10    U ← v_max - v_min;
11    Img_Bin(n, :) ← Binarize
      Img_Laser(n, :) under the threshold U;
12    Img_Bin_2(n, :) ← Perform medium
      filtering at Img_Bin(n, :);
13    Img_Skel(n, :) ← Skeletonize
      Img_Bin_2(n, :);
14    RGB_Skel(n, :) ← Extract RGB values
      from LOff(n, :);
15  end
16 end

```

is clockwise $dir = 1$, otherwise counterclockwise where $dir = -1$.

Once the rotation matrix R_p is obtained, we replace this value in the equation (5), to finally obtain the set of 3D points (X_p, Y_p, Z_p) that define the region of the object that is being inspected.

3.5. 3D Point Cloud Alignment

The alignment of 3D point clouds allows establishing the correspondences between the images of regions acquired from the test object and the ideal model of the same, which is stored in a database.

For this alignment, we need to use a 3D descriptor that can efficiently take advantage of all the information extracted from the 3D acquisition and reconstruction stages. The *multiscale covariance descriptor* (MCOV) [30] implements improvements in the calculation of its performance by incorporating several neighborhood magnitudes for each central point that is analyzed. Furthermore, this descriptor is complemented by a game theory approach called evolutionary stable strategy (ESS) to solve the coincident correspondences under global geometric constraints. This layer offers a complete understanding of the scene and avoids possible misalignments due to repeated areas or symmetries, which would be impossible to identify solely by a detector at a local level.

To select those points that can describe the object, this descriptor uses covariance matrixes whose dimensions depend directly on the number of characteristics used. The descriptor uses six characteristics, which can be extracted from a point cloud; the rotational invariant angular measures α , β and γ along with the color space values R , G and B .

Subsequently, the distance matrix between the descriptors is calculated to obtain a set of candidate pairs. Using the ESS solver, the previously obtained candidates are discarded until the best pairs are found through a defined heuristic function. Finally, with the classified pairs, the rigid transformations (rotation matrix R and translation vector t) are obtained to align the point clouds. See more theoretical details of the MCOV descriptor and its use to align 3D point clouds in [30].

The alignment stage is defined as follows: before starting and as a prerequisite it is necessary to unify the coordinates of each point with their respective values RGB . Then, as inputs, the acquired region of the test object, (Target Object: $TObj$), and the ideal model of the object (Source Object: $SObj$) are used, in order to obtain the rotation matrix, R and the translation vector t , to transform the base object and align it with respect to the real object, as shown in figure 4.

3.6. Damage Detection

With both point clouds aligned ($SObj$ y $TObj$), we can start the damage detection. This stage is important in our methodology because it is where the comparison between $SObj$ and $TObj$ is made to detect the existence -or not- of any difference, which, if it exists, we call it “damage” present in the test object.

Having said this, it is necessary to detail the procedure defined to fulfill the purpose of this stage; we begin by per-

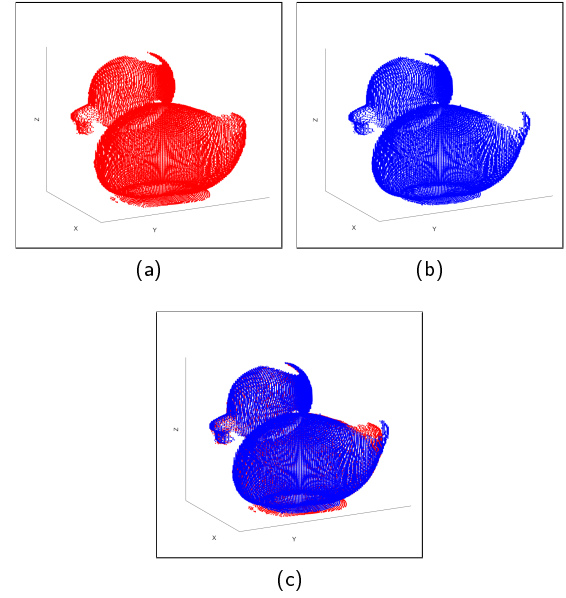


Figure 4: Objects alignment. (a) Source Object $SObj$, (b) Target Object $TObj$, (c) Source and Target Object aligned.

Algorithm 4: Function Alignment

Input: $TObj$ %3D Reconstruction of the target object, $SObj$ %3D Reconstruction of the source object.

Output: Al_SObj %Aligned 3D Reconstruction of the source object.

```

1  $Al\_SObj \leftarrow \text{Alignment}(TObj, SObj)$ 
2    $[R, t] \leftarrow \text{Descriptor\_MCOV}(TObj, SObj);$ 
3    $Al\_SObj \leftarrow \text{Align } SObj \text{ with } TObj \text{ using } R$ 
   and  $t;$ 
4 end
```

forming a search of all pairs of equivalent points between $TObj$ and $SObj$. That is, all the points between both point clouds that can be defined as similar in terms of spatial location. To do this, we use the classifier, k -NN, *i.e.*, k nearest neighbor [31], which is based on the execution of a Euclidean distance calculation [32] and subsequent analysis to find the k points that have the smallest distance value.

The calculation of the Euclidean distance d between two points p_t and p_s is done as follows:

$$d(p_t, p_s) = \sqrt{(x_t - x_s)^2 + (y_t - y_s)^2 + (z_t - z_s)^2} \quad (8)$$

where, x_t , y_t and z_t are the coordinates of a p_t point belonging to $TObj$, and x_s , y_s and z_s are the coordinates of a p_s point located at $SObj$, both with respect to the axes X , Y and Z .

The calculation of $d(p_t, p_s)$ is performed to obtain the distances between p_t and all the points that make up $SObj$ and then select the k points with the lower values, k being the number of neighboring points that we want to find. For

our method, we have adjusted $k = 1$. As a result of this, we get eq_pts , a vector containing all pairs of equivalent points between $SObj$ and $TObj$.

To determine if the pairs of equivalent points in eq_pts belong to a damage region, it is necessary to define a criterion, which indicates whether a pair of equivalent points is considered as “damage” -or not-. For this, we use the distance value d obtained in (8) and compare it with an empirical threshold value $minDist$, which is defined as the minimum limit distance so that a pair of equivalent points can be considered as “damage” under the following condition:

If $d \geq minDist$, the pair of equivalent points *qualifies* within a damage region and is stored in the vector Dmg_Obj . Otherwise, if $d < minDist$, the pair of equivalent points *does not qualifies* and is discarded.

Since there might eventually be values that exceed the $minDist$ threshold value criteria, we apply a filter to remove any pair of equivalent points that do not truly belong to Dmg_Obj . This is achieved by analyzing the distribution of the points that make up Dmg_Obj , i.e., we look for the *stp* step number of each point in Dmg_Obj and then we analyze the frequency of steps, which we obtained in the stage of 3D reconstruction.

With the procedure described, all the points considered as “damage” are contained in Dmg_Obj . As our proposal is for *active inspection*, we need to know if the region with damage continues beyond what has been found so far, i.e., if further inspection of the test object is required. For this, from Dmg_Obj we obtain its volume limit values and thus compare them with the limit values of T_Obj . In this way, when these limits are close, a new rotation cycle of the rotating platform is carried out, until fully completing the damage.

3.7. Volume Calculation

When all the inspection of a region with damage has been carried out, i.e., no proximity of limits is detected; we begin with the volume calculation. For this, we need to transform Dmg_Obj , which is currently a point cloud, into a surface. To this end, we use *AlphaShape* [33, 34], which allows us, through polygon composition, to generate a bounding volume that surrounds the point cloud. The calculation of the volume of the damage region is obtained by calculating each polygon, as follows:

$$V_p = Length \cdot Width \cdot Height \quad (9)$$

After calculating the volume for each V_p polygon that makes up the model of the damaged region, only the total volume of damage Vol should be obtained, which would be the sum of the n polygons generated by the *AlphaShape*:

$$Vol = \sum_{i=1}^n V_p \quad (10)$$

The above procedure is better described in algorithm 5.

Algorithm 5: Function Calc_Vol

Input: Dmg_Obj %3D Reconstruction of the damage found on the target object.

Output: Vol %Damage volume.

```

1  $Vol \leftarrow Calc\_Vol(Dmg\_Obj)$ 
2    $AS\_Dmg \leftarrow$  Perform AlphaShape of
    $Dmg\_Obj$ ;
3    $n\_p \leftarrow$  Number of polygons created in
    $AS\_Dmg$ ;
4   for  $n = 1$  to  $n\_p$  do
5      $V_p(n) \leftarrow$  width of  $AS\_Dmg(p) \times$  height of
        $AS\_Dmg(p) \times$  length of  $AS\_Dmg(p)$ ;
6   end
7    $Vol \leftarrow \sum_{i=1}^n V_p$ ;
8 end
```

4. Evaluation

To validate the performance of the proposed methodology, we have implemented a system that considers everything defined above and the design shown in Figure 1, which allows us to carry out tests using different objects, which may -or may not- have some damage¹.

4.1. System Design

The components used in the damage detection system are the following: a ViewSonic brand monitor, a Dell Octiplex 7050 computer with an Intel Core i7-7000 3.60 GHz processor, an image acquisition system and a motor-laser line controller (see Figure 5).

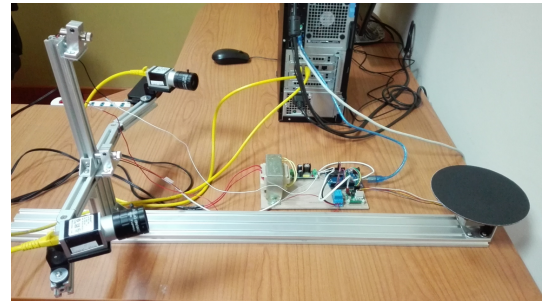


Figure 5: Damage detection system.

The computer has a built-in image acquisition card, NI PCIe-8233 Quad Port Gigabit Ethernet Interface Device, to control the simultaneous captures obtained from the cameras through 8-pin RJ-45 Jack Ethernet connectors.

The inspection system consists of a rotating platform using a 1.8° RS 440-420 stepper motor, 2 5mW adjustable laser diodes and 2 Basler acA1280-60gc cameras with Edmund

¹The codes and a restricted database with some test examples (off-line) are available at the following download link: <https://drive.google.com/drive/folders/1SAV1TA7-9I5vU7msq-TC6R2xHfV86cHa?usp=sharing>

Optics CFFL F1.3 f8.5mm lenses. The motion controller consists of an Arduino Uno board, a motor driver module, a 1-channel relay module, a voltage regulator, and a power source.

4.2. Performance Testing

We used five objects, shown in Figure 6, to perform a series of tests in three different inspection scenarios: *i*) inspection to an object without damage, *ii*) inspection to an object with additive damage, and *iii*) inspection to an object with subtractive damage. In addition, each test was repeated three times using different initial positions of the object in order to observe the number of rotations given by the system to detect and calculate the volume of damage (if any).

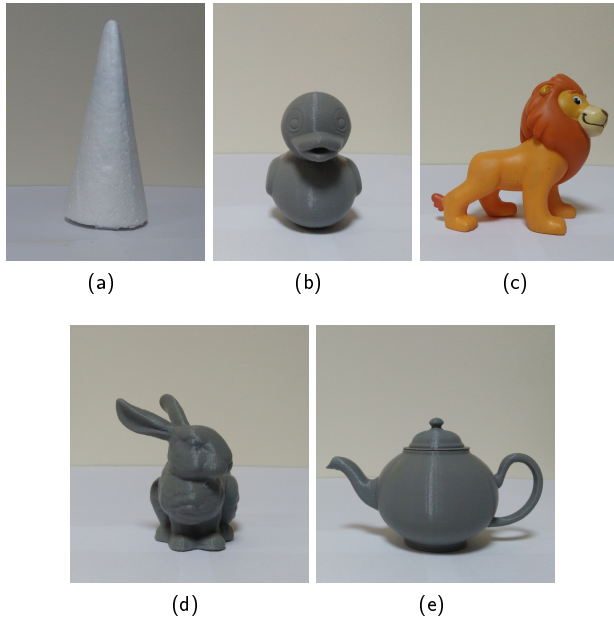


Figure 6: Target object. (a) Cone, (b) Duck, (c) Lion, (d) Stanford Bunny, and (e) Teapot.

To corroborate the volume calculation obtained, all the damages present in the objects have been created intentionally and their volume is known.

4.2.1. Inspection to Objects without Damage

The requirement to perform this type of inspection is simple: use the unaltered object, *i.e.*, the objects shown in Figure 6. We have carried out five tests of this type (one for each object) and they were called ND (No Damage).

4.2.2. Inspection to Objects with Additive Damage

To carry out these types of tests, we incorporate known volumes into the test objects, using plasticine. To check that the volume calculated with our system corresponds with the added volume, we have built three plasticine cubes to add to the five objects in a uniform way in any place of these. The cubes are 10 [mm], 15 [mm] and 20 [mm] edgewise, and therefore with volumes of 1000 [mm³], 3375 [mm³] and 8000 [mm³], respectively (see Figure 7).

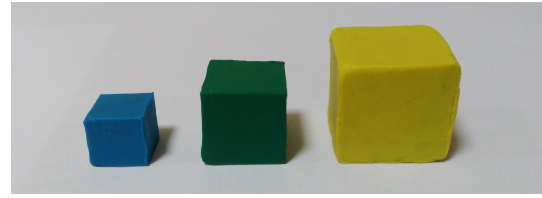


Figure 7: Plasticine cubes of 1000 [mm³], 3375 [mm³] and 8000 [mm³].

We have performed *i* inspections with additive damage for each object, named AD, *i.e.*, AD_{*i*}, where *i* = 1, 2, 3. Depending on the value of *i*, we use a plasticine cube to add to the object; If *i* = 1 we use the cube of 1000 [mm³], if *i* = 2 we use the cube of 3375 [mm³] and if *i* = 3 we use the cube of 8000 [mm³].

For each inspection test, we have added only one plasticine cube to the object, as can be seen in Figures 8a, 8b and 8c showing the object used for the Stanford Bunny AD₁, Teapot AD₂ and Cone AD₃ tests, respectively.

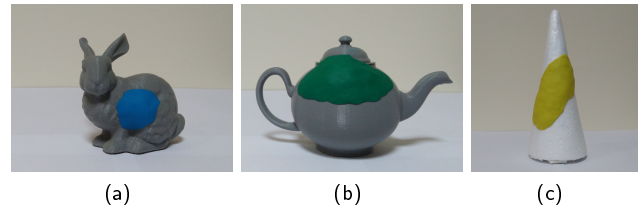


Figure 8: Objects with additive damage. (a) Stanford Bunny AD₁, (b) Teapot AD₂, and (c) Cone AD₃.

4.2.3. Inspection to Objects with Subtractive Damage

For this type of inspection, we have duplicated some of the objects (Duck, Stanford Bunny and Teapot) using 3D printing with small differences, which were eliminated to consider it as damage. In the case of the expanded polystyrene cone, we have altered it manually for the inspection.

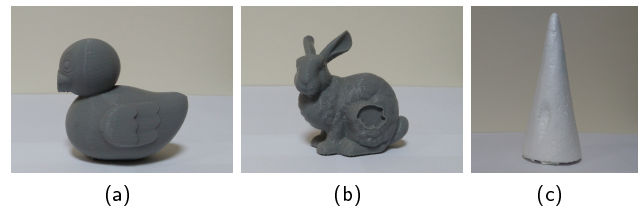


Figure 9: Objects with subtractive damage. (a) Duck SD₁, (b) Stanford Bunny SD₂, and (c) Cone SD₃.

We have performed *i* inspections with subtractive damage for each object, named SD, *i.e.*, SD_{*i*}, where *i* = 1, 2, 3. In Figure 9a, 9b and 9c the objects used to perform the Duck SD₁, Stanford Bunny SD₂ and Cone SD₃ tests are shown respectively.

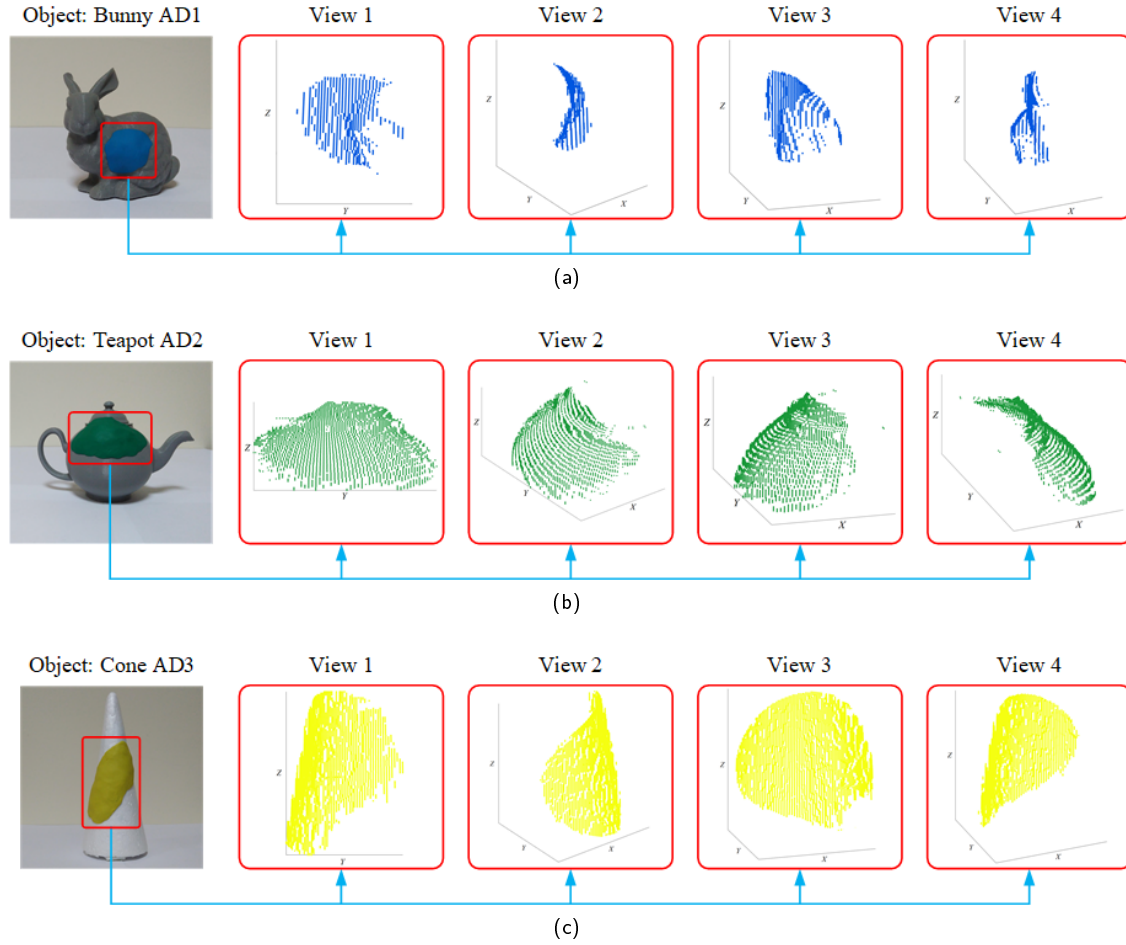


Figure 10: Additive damage 3D reconstruction for testing: (a) Stanford Bunny AD1, (b) Teapot AD2, and (c) Cone AD3

4.3. Analysis of Results

We have carried out a total of 84 inspections, 15 to objects without damage, 45 to objects with additive damage and 24 to objects with subtractive damage. We can see in Figures 10 and 11, some views of the 3D reconstructions of the damage, obtained from the tests carried out on the objects shown in Figures 8 and 9 respectively.

To evaluate the performance of the proposed methodology, we have used the *Percentage error* ($error$ [%]) and *Mean Absolute Percentage Error* ($MAPE$). In addition, we count the rotations necessary to achieve the detection of possible damage. These results are shown in Table 1.

The $error$ [%] is calculated using the following equation (11):

$$error[\%] = \frac{|Vol_r - Vol|}{Vol_r} \cdot 100 \quad (11)$$

where, Vol_r is the actual volume of damage, and Vol is the estimated volumen of damage calculated with our method.

To determine the global error of our proposal in the tests carried out, we must calculate the *Mean Absolute Percentage*

Error ($MAPE$) using the equation (12):

$$MAPE = \frac{1}{n} \sum_{i=1}^n error[\%]_i \quad (12)$$

where, n is the number of tests performed.

From the data shown in Table 1 we can indicate that 21 tests were performed for both the Cone object and the Stanford Bunny object; 3 were ND (No Damage), 9 were AD (Additive Damage) and 9 were SD (Subtractive Damage). In addition, 15 trials were performed for the Duck and Teapot objects, 3 of which were ND, 9 were AD, and 3 were SD. For the Leon object, only 12 tests were performed, 3 ND and 9 AD. We see that the percentage errors obtained for the Duck AD1 test obtained the highest value of 12.70%, followed by the Lion AD1 test, which obtained a percentage error equal to 11%. Without considering the results obtained in the ND tests of each object, the best value was obtained by the Teapot SD3 test with 0.14% which corresponds to 11 [mm^3] difference between the real volume Vol_r and the estimated volume Vol . In relation to the rotations made to detect and quantify damage in this active inspection proposal, they range from 1 to 11 rotations, which were necessary to complete the object inspection process. It should be noted that

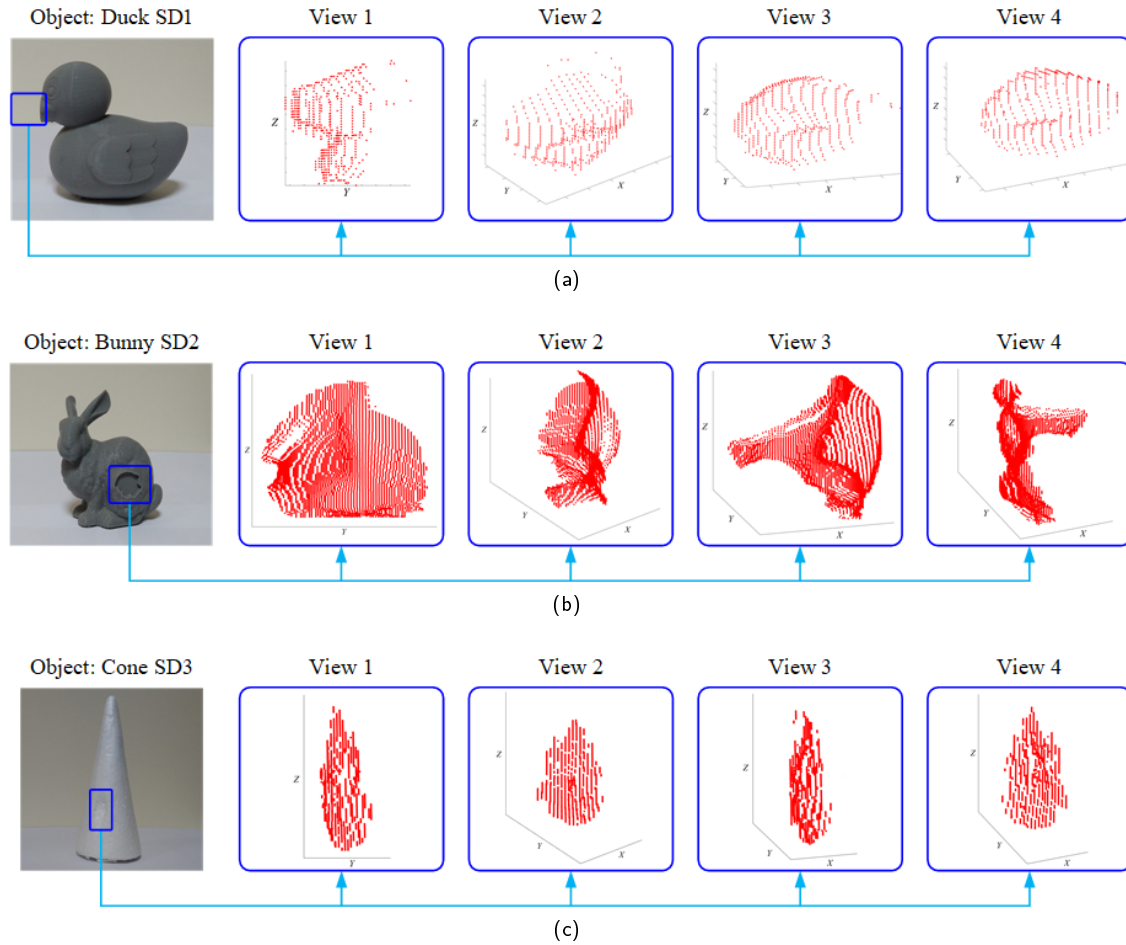


Figure 11: Subtractive damage 3D reconstruction for testing: (a) Duck SD1, (b) Stanford Bunny SD2, and (c) Cone SD3.

in all the ND tests a total of 6 rotations were performed to complete the inspection and ensure that there was no damage.

Finally, the inspection and damage calculation system yielded a value of $MAPE = 2.78\%$, so we can say that our proposal allows for a very efficient inspection of objects with calculated damage volumes close to the real values.

Furthermore, in Table 1 we can see that for tests where the test object has existing damage, the system is capable of executing an adequate number of rotations to be able to reconstruct and calculate the volume of damage detected, reducing the steps to complete the inspection.

The main restriction of our method is that the rotation is only performed for the Y axis and we have used only two cameras, which allow acquiring the deformation of the laser lines. There are areas of the test object that cannot be reconstructed, and consequently, no damage (if any) can be detected. This can occur at the top and bottom ends of the object and in intricate areas, where the laser line is not well defined or simply because the laser line fails to intersect with the surface of the object. This can be avoided by performing two additional inspections of the object; one by rotating the object 90° and one with the object rotated to -90° , both for

the X axis. Another solution to this problem could be to incorporate more laser lines and cameras, which would allow inspecting more areas that are difficult.

5. Conclusions

In this work, we implement a methodology to detect damage present in objects using an approach based on active search. The benefit it presents lies in its effectiveness when inspecting a target object since it performs it by sections (defined amount of captures obtained for each rotation given by the rotating platform) until it finds an incongruous area compared to a model of the object without damage, stored in a database.

To check the performance of the inspection system, we have carried out 84 tests, in three case scenarios: *i*) inspection of an object without damage, *ii*) inspection of an object with additive damage, and *iii*) inspection of an object with subtractive damage. The results we have obtained are auspicious and indicate that the system we propose has a *Mean Absolute Percentage Error* of $MAPE = 2.78\%$, confirming that the system presents an acceptable inspection analysis.

The biggest *Percentage error* ($error[\%]$) has been ob-

Table 1
Performance testing for damage detection system.

Object	Test	No. of rotations to complete the test			Damage volume real, [mm ³] <i>Vol_r</i>	Damage volume estimated, [mm ³] <i>Vol_e</i>	error[%]
		Obj 1	Obj 2	Obj 3			
Cone	ND	6	6	6	0	0	0.00
	AD1	5	5	6	1000	959	4.10
	AD2	3	3	7	3375	3173	5.99
	AD3	7	9	8	8000	7565	5.44
	SD1	11	11	11	4570	4350	4.81
	SD2	6	4	5	2845	2640	7.21
	SD3	1	1	4	3266	3055	6.46
Duck	ND	6	6	6	0	0	0.00
	AD1	4	2	2	1000	873	12.70
	AD2	4	5	3	3375	3166	6.19
	AD3	7	9	9	8000	7781	2.74
	SD1	1	3	5	9744	9558	1.91
Lion	ND	6	6	6	0	0	0.00
	AD1	1	3	5	1000	890	11.00
	AD2	1	1	1	3375	3356	0.56
	AD3	1	6	7	8000	7981	0.24
Stanford Bunny	ND	6	6	6	0	0	0.00
	AD1	1	2	7	1000	950	5.00
	AD2	1	8	4	3375	3356	0.56
	AD3	1	8	10	8000	7981	0.24
	SD1	1	1	1	2210	2200	0.45
	SD2	1	4	4	3450	3420	0.87
	SD3	1	2	7	4200	4188	0.29
Teapot	ND	6	6	6	0	0	0.00
	AD1	1	11	11	1000	995	0.50
	AD2	1	5	7	3375	3369	0.18
	AD3	1	4	4	8000	7989	0.14
	SD1	1	11	11	9527	9500	0.28
MAPE =							2.78

tained from objects with additive damage of size 1000 [mm³], and this can be caused by the section where said damage was incorporated. It could be an intricate area, where the laser line is not well defined and/or the position of the cameras prevents the total visualization of the laser line.

Using an efficient combination of machine vision algorithms, we have proposed a method that leaves static inspection methods behind, introducing the concept of active vision for damage search. In addition, traditional inspection methods with machine vision algorithms only go as far as 3D reconstruction and damage detection; however, they do not incorporate automatic damage measurement. With our method, we have demonstrated that it is possible to measure the volume of surface damage of an object, sufficiently close to the real value.

It should be noted that this system is a first approach to this methodology and observing the results obtained we think that some relevant aspects were not considered, such as the inspection time and the sections of the object not analyzed —therefore not reconstructed— due to the limited number of cameras, among other aspects. As future work, we propose improving the image acquisition structure so that it allows acquiring images that include sections that were currently not fully observed and consequently not reconstructed in 3D, *e.g.*, lower and upper parts of the object. In other words, increasing the number of cameras and giving them greater freedom of positioning for greater inspection coverage.

Acknowledgements

This work was supported in part by DIUDA Grant No. 22406 and No. 22345 from Universidad de Atacama.

References

- [1] J. See, C. Drury, A. Speed, A. Williams, and N. Khalandi, "The role of visual inspection in the 21 st century," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 61, pp. 262–266, 2017.
- [2] T. Czimmermann, G. Ciuti, M. Milazzo, M. Chiurazzi, S. Roccella, C. Oddo, and P. Dario, "Visual-based defect detection and classification approaches for industrial applications -A survey," *Sensors*, vol. 20, p. 1459, 2020.
- [3] V. Rizzo and D. Mery, "Automated detection of threat objects using adapted implicit shape model," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, pp. 472–482, 2016.
- [4] N. Vundekode, P. Kalapatapu, and V. Pasupuleti, *A Study on Vision Based Method for Damage Detection in Structures*, pp. 96–105. 01 2021.
- [5] G. Nikonova, A. Nikonov, Y. Zakirova, and V. Makarochkin, "Automated system for visual non-destructive testing," pp. 1–5, 2021.
- [6] M. Isa and I. Lazoglu, "Design and analysis of a 3d laser scanner," *Measurement*, vol. 111, 2017.
- [7] Y. Liu, Y. Wang, X. Cai, and X. Hu, "The detection effect of pavement 3D texture morphology using improved binocular reconstruction algorithm with laser line constraint," *Measurement*, vol. 157, p. 107638, 2020.
- [8] Y. Wang, N. Deng, and B. Xin, "Investigation of 3d surface profile reconstruction technology for automatic evaluation of fabric smoothness appearance," *Measurement*, vol. 166, p. 108264, 2020.
- [9] Z. Tian, L. Jianhua, L. Shaoli, T. Chengtong, and J. Peng, "A 3D reconstruction method for pipeline inspection based on multi-vision," *Measurement*, vol. 98, 2016.
- [10] J. Molleda, R. Usamentiaga, F. D. Garcia, F. Bulnes, and L. Ema, "Shape measurement of steel strips using a laser-based three-

- dimensional reconstruction technique,” *IEEE Transactions on Industry Applications - IEEE TRANS IND APPL*, vol. 47, pp. 1–8, 2010.
- [11] J. Molleda, R. Usamentiaga, F. D. Garcia, F. Bulnes, A. Espina, B. Di-eye, and L. Smith, “An improved 3D imaging system for dimensional quality inspection of rolled products in the metal industry,” *Computers in Industry*, vol. 64, pp. 1186–1200, 2013.
 - [12] H. Kwon, M. Na, and J.-B. Song, “Rescan strategy for time efficient view and path planning in automated inspection system,” *International Journal of Precision Engineering and Manufacturing*, vol. 20, pp. 1747–1756, 2019.
 - [13] A. Dawda, M. Nguyen, and R. Klette, “Accurate 3d measurement of highly specular surface using laser and stereo reconstruction,” pp. 1–6, 12 2019.
 - [14] A. Dawda and M. Nguyen, “Defects detection in highly specular surface using a combination of stereo and laser reconstruction,” pp. 1–6, 11 2020.
 - [15] F. Zidane, J. Lanteri, C. Migliaccio, and J. Marot, “System measurement optimized for damages detection in fruit,” in *2021 IEEE Conference on Antenna Measurements Applications (CAMA)*, pp. 550–554, 2021.
 - [16] B. Güldür Erkal and J. Hajjar, “Using extracted member properties for laser-based surface damage detection and quantification,” *Structural Control and Health Monitoring*, vol. 27, 07 2020.
 - [17] B. Zhang, N. Guo, J. Huang, B. Gu, and J. Zhou, “Computer vision estimation of the volume and weight of apples by using 3d reconstruction and noncontact measuring methods,” *Journal of Sensors*, vol. 2020, pp. 1–12, 11 2020.
 - [18] C. Liu, L. Zhou, W. Wang, and X. Zhao, “Concrete surface damage volume measurement based on three-dimensional reconstruction by smartphones,” *IEEE Sensors Journal*, vol. 21, no. 10, pp. 11349–11360, 2021.
 - [19] S. Mitra, S. K. Pal, and P. Mitra, “Data mining in soft computing framework: a survey,” *IEEE Transactions on Neural Networks*, vol. 13, pp. 3–14, 2002.
 - [20] A. Kus, “Implementation of 3D optical scanning technology for automotive applications,” *Sensors (Basel, Switzerland)*, vol. 9, pp. 1967–1979, 2009.
 - [21] M. Gupta, M. Khan, R. Butola, and R. Singari, “Advances in applications of non-destructive testing (NDT): A review,” *Advances in Materials and Processing Technologies*, pp. 1–22, 04 2021.
 - [22] V. K. Mishra, S. Kumar, and N. Shukla, “Image acquisition and techniques to perform image acquisition,” *SAMRIDDHI : A Journal of Physical Sciences, Engineering and Technology*, vol. 9, 2017.
 - [23] M.-A. Drouin and J.-A. Beraldin, *Active Triangulation 3D Imaging Systems for Industrial Inspection*, pp. 109–165. 09 2020.
 - [24] J. Cock, “The triangulation method applied in a laser scanner, for three-dimensional objects,” *EAFIT University Magazine [in Spanish]*, vol. 36, pp. 25–31, 2012.
 - [25] D. Mayorca and A. López Rubio, “Reconstruction of 3D objects with axial symmetry from laser triangulation,” *UNIMAR Magazine [in Spanish]*, vol. 35, pp. 239–253, 2018.
 - [26] L. Long and S. Dongri, *Review of Camera Calibration Algorithms*, pp. 723–732. 05 2019.
 - [27] Z. Zhang, “A flexible new technique for camera calibration,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, pp. 1330–1334, 2000.
 - [28] R. Gonzalez and R. Woods, *Digital Image Processing*. Pearson, 4th ed., 2018.
 - [29] V. Tyagi, *Understanding Digital Image Processing*. 09 2018.
 - [30] P. Cirujeda, Y. Dicente Cid, X. Mateo, and X. Binefa, “A 3D scene registration method via covariance descriptors and an evolutionary stable strategy game theory solver,” *Int. J. Comput. Vision*, vol. 115, pp. 306–329, 2015.
 - [31] H. Wang, I. Dntsch, and G. Gediga, “Nearest neighbours without k: A classification formalism based on probability,” 2003.
 - [32] J. M. Phillips, “Distances and nearest neighbors,” in *Mathematical Foundations for Data Analysis*, pp. 59–94, Springer, 2021.
 - [33] J. Gardiner, J. Behnsen, and C. Brassey, “Alpha shapes: determining 3D shape complexity across morphologically diverse structures,” *BMC Evolutionary Biology*, vol. 18, 12 2018.
 - [34] Z. Liao, J. Liu, G. Shi, and J. Meng, “Grid partition variable step alpha shapes algorithm,” *Mathematical Problems in Engineering*, vol. 2021, pp. 1–8, 05 2021.