

A Study of Reverse Engineering Program Based on Obtaining Point Clouds from Object Images

Gürcan Samtaş

Faculty of Engg., Department of Mechatronics Engg., Düzce University, Düzce,-Türkiye

Abstract: *This study focuses on developing a program which obtains 3D point clouds by interpreting images taken with a digital camera, using image processing. In the developed program, images taken with a digital camera without using a specific shooting condition are evaluated pixel by pixel. These evaluations are made with reference points marked on the image in the designed area by user. The 3D point clouds of the objects in the marked area can be obtained in a short time interpreting pixel values of the limited area. In the study, a different approach is presented, in which image processing and Reverse Engineering (RE) techniques are used, in the process of 3D design and manufacturing. Furthermore, the developed program has the advantages of keeping a relatively short processing, and because of using reference area, images and point clouds used can express objects in image. In this study, in addition, algorithms used while processing images in developed program are explained on a sample image in detail, and sample images which obtained point clouds and their solid models are given.*

Keywords: *reverse engineering; CAD; image processing; point cloud*

I. INTRODUCTION

Engineering is a process conducting activities related to design, analysis, manufacturing and maintenance of the system and presenting analytical solutions to problems by modeling, physical events, and situations mathematically. Forward engineering and reverse engineering are two types of engineering. Forward engineering is the traditional process of moving from high-level abstractions and logical designs to the physical implementation of a system. In some situations, there may be a physical part/product without any technical details, such as drawings, bills-of-material, or without engineering data. The process of duplicating an existing part, subassembly, or product, without drawings, documentation, or a computer model is known as reverse engineering [1]. In other words, reverse engineering can also be defined as the process of digitizing existing parts and acquiring a Computer Aided Design (CAD) model with three dimensional (3D) point clouds. Reverse engineering, which is defined as a process of digital capturing of one element's physical existence, can be defined in different ways according to special business areas by researchers [2].

The development of image processing applications has proven to be a highly complex and time consuming activity. This prevents the penetration of image processing into industry, whereas the demand for concrete applications is growing in all domains: medicine, biotechnology, remote sensing, astronomy, ecology, security, forensics, chemistry, surveillance, quality control, etc. As a result, it is becoming indispensable to develop systems that can help end-users build customized applications. The challenge is to allow end-users to program software at the image feature level rather than at the computer code level [3]. In image processing main goals are to obtain useful information about the scene and to observe the image in a good way by defining the image. Here, machine or human is the observer. If raw material is not clear enough, it should be treated for the desired purpose. When transactions are often done by people in an electronic environment, optical and electronic devices are needed to obtain the image. When digital image processing is applied in accordance with the purpose, it has many advantages such as speed, low cost and achieving the desired result. There are also architectural applications in the RE area. A different approach has developed for image based architectural modeling, which processes images, by using library support of 3D objects. With the presented model, the images which are 1024x768, 24 bit and BMP extension are used. In the system, a model library is used to generate realistic 3D models [4].

Primitive recognition methods are used in image processing area which uses model mapping comment rate in computer aided design and photometry measurement systems, and this study consists of review angle, graphic angle and model matching [5]. Three dimensional models of urban areas are generated with the help of such devices as 2D laser beam scanner, density cameras. There are several approaches to obtain 3D models of cities, for instance the objects in images obtained by satellite or antenna are determined by stereo vision or synthetic aperture radar stereo. In this approach, although the objects are generated quickly, the resolution and accuracy of generated models are also very low [6, 7]. On the other hand, 3D camera tracking systems are used occasionally in industrial augmented reality applications. Bleser et al. [8] presented real time 3D camera tracking systems, and with this presented solution, there is no need for forecasting, posing, monitoring systems and pointers used in other similar applications. Augmented reality offers new approaches such areas as machine repair, design, medicine and cultural heritage [9-11]. Song and Wang [12] have developed a new grid matching method for 3D reconstruction. Three dimensional automatic building of surfaces has been the subject of research of digital photometry for many years. Today, photometry is applied in the fields of both computer vision and computer graphics such as industrial inspection and reverse engineering [13, 14]. The reconstruction of 3D models includes four main steps such as data capture, recording, surface integration and texture map. Partially generated 3D shapes and texture information are obtained from a different point of views [15]. Computer vision and image understanding are areas of research related to incremental number of images, image quality, object perception in the image and its applications [16, 17]. Barbero and Ureta

[18] compared five digitization techniques in the field of Reverse Engineering for the quality of the distribution of points and triangular meshes: 1. An ordered point cloud obtained with a laser incorporated in a Coordinate Measurement Machine (CMM); 2. A disordered point cloud obtained with a manual laser, the position of which is determined with a Krypton Camera; 3. An EXA scan manual laser with targets; 4. An ordered point cloud obtained by high accuracy computerized tomography; 5. An atos fringe projection scanner with targets. Bosche [19] presented a new approach for automated recognition of project 3D Computer-Aided Design (CAD) model objects in large laser scans. Test results were given with data obtained from the construction of an industrial building's steel structure. Yang et al. [20], in their study, presented a fast and precise 3D laser scanner to recover point cloud data and then import the data into Rapid Form XOR2, an application used in reverse engineering, to process point cloud data and construct 3D models. Their study results were converted from 3D models into 2D pictures by parameterization software Pro/ENGINEER to provide designers precise and realistic dimensional data, which efficiently reduced time in cartography and increased historic building dimension precision during digitization.

In this study, it is aimed to develop a RE application which processes color images obtained by a digital camera, achieves 3D point clouds from these images without using a specific shooting environment. In the developed program, .NET programming sub-structure and PHP scripting language have been used, and 3D point clouds are obtained by processing color images belonging to sample engineering models. Three dimensional point clouds obtained from the program are converted into solid models in the CATIA design program to obtain reconstruction of 3D objects.

II. EXPERIMENTAL WORKS AND IMAGE ANALYSIS

Experimental work is done to testing of the program and developing of the image processing algorithms. The obtained samples images with the prepared testing apparatus are used in developing program (fig.1). Sample models are expressed in two views. Images are obtained from a general environment without the need for a specific shooting condition. In developed program, jpg and bmp formatted images taken from a digital camera can be used. To obtain pixel values in the images, RGB (red, green, blue) color system is used. In RGB color value, each color has maximum 256 color value. Combination of three colors (256x256x256) consists of a total of 16,777,216 colors. These colors are expressed as real colors. For example, a pixel having (178, 229, 41) colors is expressed as (10110010, 11100101, 00101001) in the binary system. These values are defined for each pixel. Pixels on the processed images in the system are scanned according to color values. During this scanning, pixels of the object are detected according to the color values and translated into the language of binary encoding and stored temporarily side by side as 0 and 1 without distinction. When colors are scanned pixel by pixel, each color values up to +12 or -12 is considered the same fig. (2).

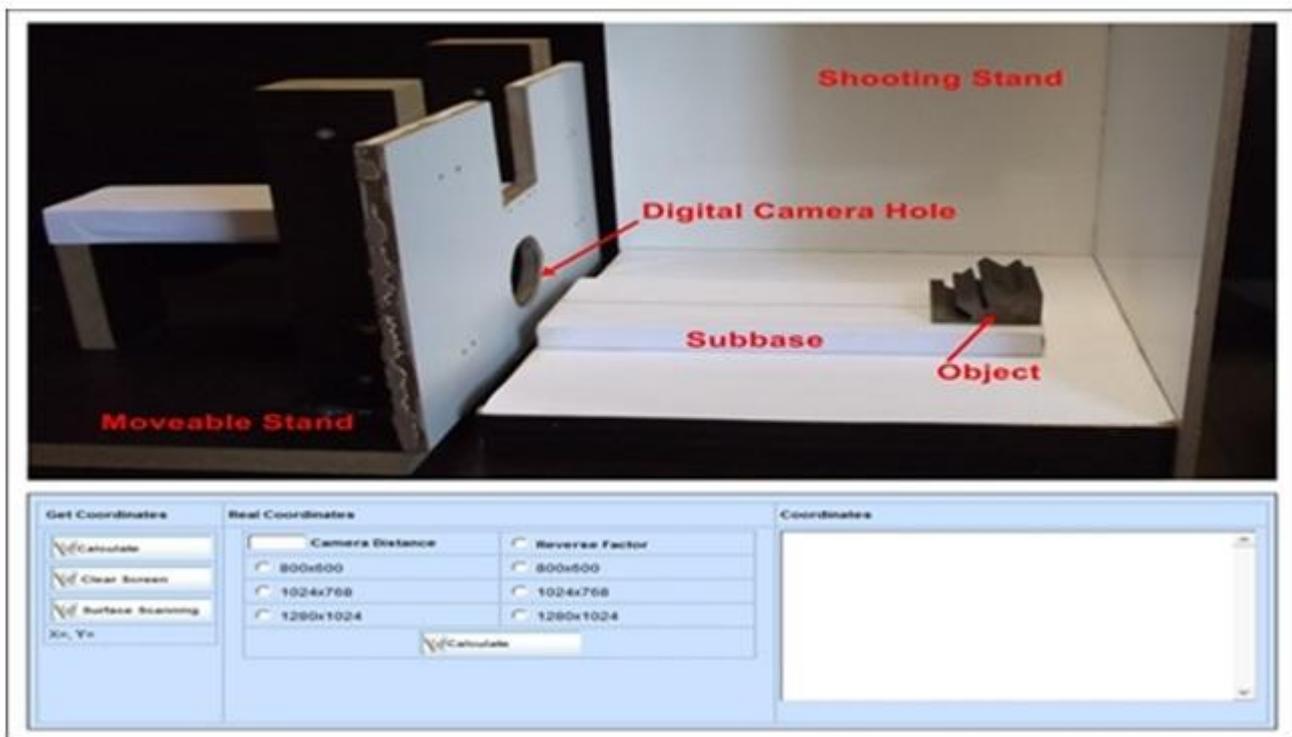


Figure1: Testing apparatus and developed system interface

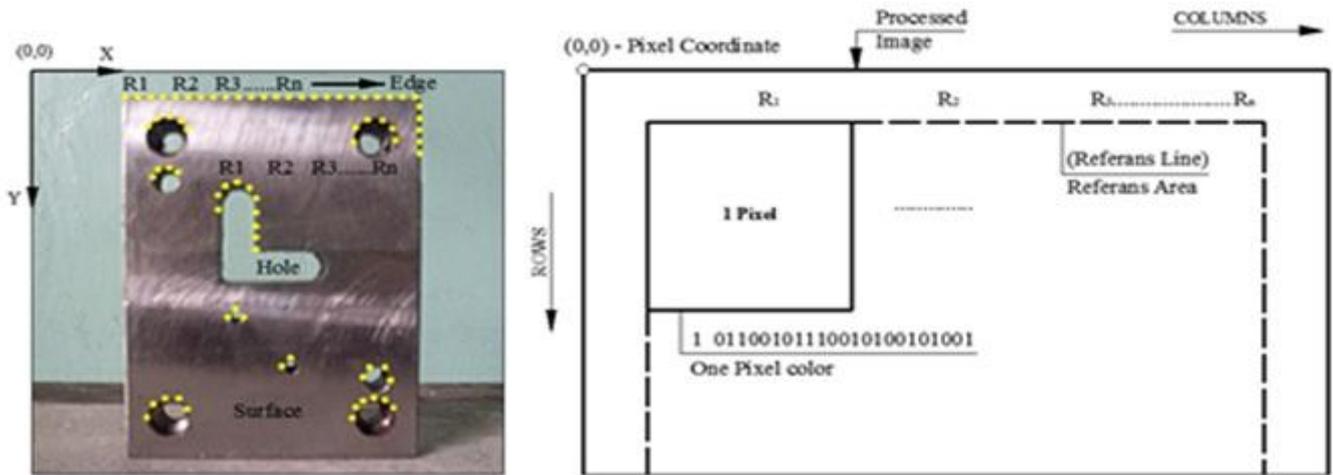


Figure 2: Color analysis applied to an image

Where R1 R2 R3 ... Rn are reference points, and they marked by user. Images are processed according to color values, evaluating the whole image requires long processing time. Therefore, as shown fig. (2), because of reference points are given to target object in the image, the processing time is shortened. This process will also help to identify the edges of the object, and system scans from left to the right in the reference area, starting from the first marked reference point.

III. OBTAINING POINT CLOUDS

Reference points are marked from right to the edges of the target object in the image randomly (fig. 2). As a result of the analysis, the scanning for obtaining point clouds is superficial, and slopes on the surface are determined according to perpendicular orientations. These perpendicular orientations are horizontal or vertical orientations, which are according to X and Y planes, on the front view of the part. Therefore, circular edges of processed piece can be processed as superficial. In the system, to interpret the images, edge-hole model and surface-area model have been developed in two different algorithms.

Edge-Hole Model

Edge-hole model algorithm is based on marking surface and hole-edges of the processed piece. It is a model which is created for edge and hole of the processed piece. In general, there are edges of each object, but holes may not be available in all of them. In this model, holes are processed as edges. After making the reference point, and the color analysis is applied to the area within the marked points. If there is an area which will be processed as a hole in the outer surface area, this time program scans the area which is between these two markings and applies color analysis to two different areas (fig. 3).

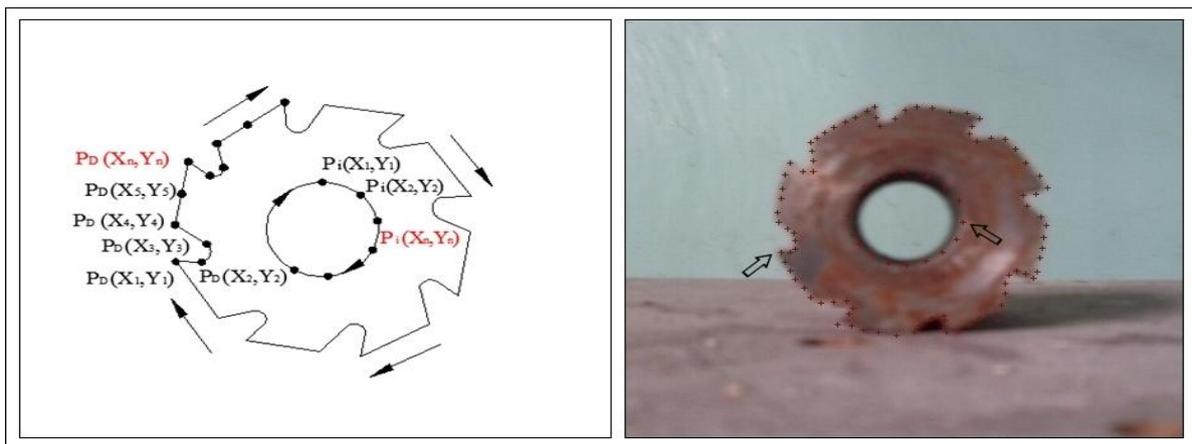


Figure 3: The processed image and the edge-hole model algorithm applied to the image

The processed sample image and edge-hole model algorithm applied by the system and the marked reference points to this image are shown in fig. (3). Reference points are marked randomly to the edges and holes of the target object in the image by the user. The system combines the $P_D (X_I, Y_I)$ starting point marked for outer surface with the ending point $P_D (X_n, Y_n)$ and scans the space which is the result of this combination.. If there is a hole in the object to be processed, as in the outer edge of the object, a $P_i (X_I, Y_I)$ starting point and an ending point $P_i (X_n, Y_n)$ are marked and a scanning in itself is carried out. The system scans the area between edge-hole and inner hole area for outer surface for this type pieces. In edge-hole model, a process applied to the images to detect the points is shown in fig. (4).

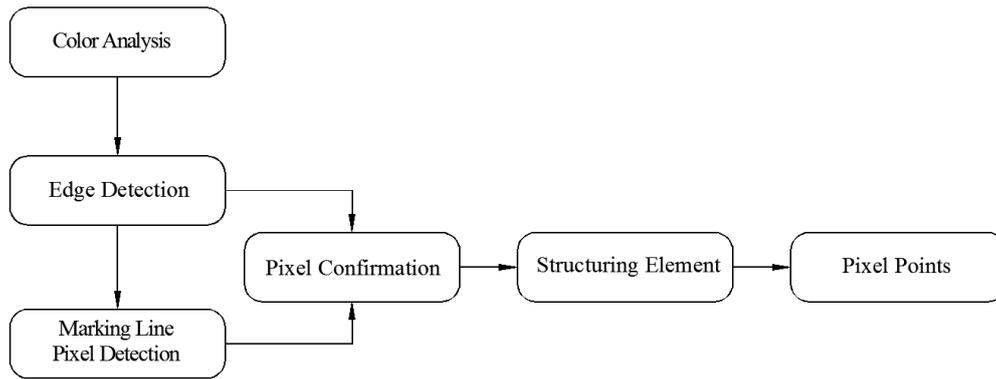


Figure 4: Processes applied to the images in the edge-hole model

In fig. (4), the processes applied to the image for edge-hole model are shown respectively. Processes applied to the image consist of 6 stages as follows: (1) providing borders with making color analysis in the inner region of the marked area; (2) determination of the edges in these provided borders with edge detection; (3) pixels detection of the line which is from the beginning to the ending of the marking according to the colors; (4) confirmation of the pixels obtained by edge detection and color analysis (5) applying structural elements to the detected edge pixels; (6) point detection from the remaining pixels. The pixels which are in the marked space are detected by the color analysis and approximate edge border is determined. With the edge detection carried out in the determined area, pixel sequence of first order derivative edge border of processed object is determined. There are several algorithms used for edge detection. Two fundamental methods are used for first order derivative edge detection from these algorithms [21]. In this study, orthogonal gradient generation method for first order derivative edge detection is used (Eq.1).

$$\theta(x, y) = \tan^{-1} \left\{ \frac{I_C(x, y)}{I_R(x, y)} \right\} \quad (1)$$

$$I_C(x, y) = P(x, y) - P(x+1, y) \quad (2)$$

$$I_R(x, y) = P(x, y) - P(x, y-1) \quad (3)$$

In Eq. 1, each column elements taken from the image and slope angle (θ) obtained from the row element is shown. In Eq. 2, column matrix obtained from pixel values in columns is shown. Each value of column matrix (IC) is defined as the difference between the pixel value in the scanning (P) and the value of the next pixel. In Eq. 3, row matrix (IR) is defined as the difference between the pixel value in the scanning and the previous pixel value [21]. The pixel matrixes obtained after the edge detection are shown as PKTd and PKTi matrices.

$$P_{KTd} = \begin{bmatrix} P_{Kd}(X_1, Y_1) \\ P_{Kd}(X_2, Y_2) \\ \dots \\ \dots \end{bmatrix}, \quad P_{KTi} = \begin{bmatrix} P_{Ki}(X_1, Y_1) \\ P_{Ki}(X_2, Y_2) \\ \dots \\ \dots \end{bmatrix}$$

The size of $nx1$ edge matrix (P_{KTd}) obtained from edge detection and the size of $nx1$ hole matrix (P_{KTi}) done for edge-hole are shown in Eq. 4. Where, P_{Kd} represents the pixel points obtained by edge detection for the outer edge and P_{ki} represents the pixel points obtained by inner edge detection. Edge pixels, which are obtained by edge detection from the outer and inner edges, are stored in the system to compare with the pixel values which will be obtained along with the line of reference point. Other groups of pixel are obtained along with the line beginning and the end of the reference points. This line is applied for the inner and outer edges as in the edge detection. The application is done the way that each marked point follows the next point along with the observed line and obtained pixel values are stored in the matrix (P_{KHd}, P_{KHl}).

$$P_{KHd} = \begin{bmatrix} P_{Hd}(x_1, y_1) \\ P_{Hd}(x_2, y_2) \\ \dots \\ \dots \end{bmatrix}, \quad P_{KHl} = \begin{bmatrix} P_{Hi}(x_1, y_1) \\ P_{Hi}(x_2, y_2) \\ \dots \\ \dots \end{bmatrix}$$

Where, P_{Hj} is expressed as inner edge pixel point marked on the inner hole line. The pixels, which are obtained through reference points marked from the outer and inner edges, are stored in the system temporarily as in the edge detection. Other pixels, which are not detected by edge detection, are covered with line detection and the pixels, which are on the slope of edge aren't detected by line detection, are detected by edge detection. The pixels detected for outer and inner edges by edge detection are compared with the pixels detected for outer and inner edges along the line. When this comparison is done with each other line by line, pixel verification is finished. Missing pixels are covered with this verification. As a result of this integration, the starting point and end point, two types of pixel appear along with the line as inner and outer as shown in the following matrices.

$$P_d = \begin{bmatrix} P_d(x_1, y_1) \\ P_d(x_2, y_2) \\ \dots \\ \dots \end{bmatrix}, \quad P_i = \begin{bmatrix} P_i(x_1, y_1) \\ P_i(x_2, y_2) \\ \dots \\ \dots \end{bmatrix}$$

Where, P_d is expressed as the combination of the pixel obtained from the edges along with the edge detection and edge line, P_i is expressed as the combination of the pixels obtained from the inner edges along with the edge detection and edge line. New pixel points, which are attained by combining the pixels obtained by edge detection and line detection, contains unwanted pixel points due to their brightness differences arising from color and light in the image. The parts containing pixel points which are unwanted and which have gone out of the line cause disturbances on the surface of the work piece in part modeling. To prevent this kind of deformation is necessary to eliminate such pixels, and unwanted pixel points are removed with implementation of structural element (fig. 4). This structural element is panned from the marked starting point to the last reference point of the detected line. Panning process is applied separately to the pixels obtained from the outer edge and the pixels obtained from the inner edge in their own group. During panning process, pixels which are on the structural element's parts specified with '0' are removed and the pixels which are on the '1' are left (fig.5).

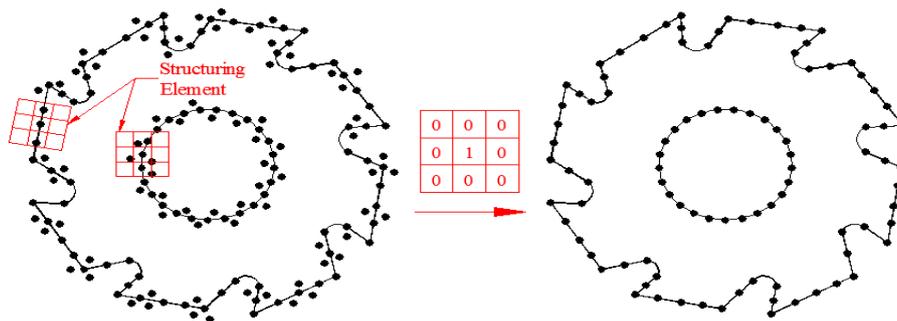


Figure 5: Implementation of the structural element

In fig (5), implementation of the structural element to the combination of the outer and inner pixels obtained by edge detection along with the reference line. With the implementation of the structural element, the last pixel line is obtained. In this way, irregularity formed by the pixels along the line is eliminated. With the elimination of the unnecessary pixels on the outer (P_d) and inner (P_i) pixel matrixes, inner pixel matrix (P_i) is added the way that it will continue under the last line element of the outer pixel matrix (P_d). With the addition of two matrixes to each other, general pixel points are obtained for edge-hole model ($P_{(X,Y)}$).

$$P_{(X,Y)} = \begin{bmatrix} P(X_1, Y_1) \\ P(X_2, Y_2) \\ \dots \\ P(X_n, Y_n) \end{bmatrix}$$

In Eq.7, pixel points obtained by applying structural element are shown in the matrix. $P_{(X,Y)}$ is the expression of combination of outer and inner pixel matrixes as one under the other. In Eq.10, X and Y values of the pixel points having X and Y values expressed lastly are obtained and coordinate matrixes are created separately ($X_{(coord)}$ and $Y_{(coord)}$).

$$X_{(coord)} = \begin{bmatrix} X_1 \\ X_2 \\ \dots \\ X_n \end{bmatrix}, \quad Y_{(coord)} = \begin{bmatrix} Y_1 \\ Y_2 \\ \dots \\ Y_n \end{bmatrix}$$

These obtained $X_{(coord)}$ and $Y_{(coord)}$ matrices are stored in the system temporarily to be combined with Z coordinates which will be obtained from the side view of the object. Two reference points obtained from the side view determine the quality of the pixel detection (fig. 6).

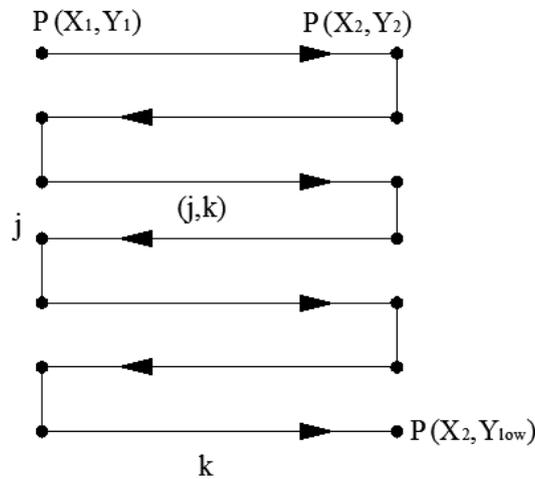


Figure- 6: Pixel detection in side view for edge and hole model

Applied pixel detection in side view for edge and hole model is shown in fig. (6). As in the front view, color analysis is performed to provide information about the edge of the part in side view. This color analysis starts from the first marked $P(X_1, Y_1)$ point, continues within the borders of the second $P(X_2, Y_2)$ point and continues to the sub-reference point of $P(X_2, Y_{Low})$ in side view. Y_{Low} is the bottom Y value of reference point detected in the front view. The pixel points which are obtained from the $(j \times k)$ sized analysis area which is determined with the marked two points and pixel points from view as reference is shown in the following matrix.

$$P_{(Z)} = \begin{bmatrix} (X_1, Y_1) & \dots & (X_2, Y_2) \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & (X_2, Y_{Low}) \end{bmatrix} \Rightarrow Y_1 = Y_2$$

The area which is between two reference points marked in the side view for edge-hole model, and the Z value of the pixels, which are detected between this area's edges detected by color analysis and represented with 'P', are determined. As in the previous pixel detections, the pixel points acquired from these processes are expressed in the form of matrix. While the scanning in horizontal is done for each X value, the scanning in the vertical is the each projection distance of the points in vertical obtained in the front view. In this way pixel points, which are acquired in vertical, have consistent and homogeneous distribution with the pixel points in the front view. Each pixel points obtained in the side view are represented with X and Y values. Each X value detected from these values is taken as Z value. In other words, each X pixel point obtained in the side view is actually Z pixel point as shown in the following matrix;

$$Z_{(coord)} = \begin{bmatrix} Z_{(j,k)} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \Rightarrow \begin{matrix} j_{first} = Y_1, & j_{end} = Y_{lower} \\ k_{first} = X_1, & k_{end} = X_2 \end{matrix}$$

This obtained Z value is combined with X and Y coordinates value obtained in the side view and converted to a single matrix form (X, Y, Z) .

$$X, Y, Z = \begin{bmatrix} X_{(n,1)} & Y_{(n,2)} & Z_{(j,k)} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ (n,1) & (n,2) & (n,3) \end{bmatrix}$$

X, Y and Z coordinate values obtained for edge-hole models are shown in the form of matrix. The points obtained in the front view as 'n', and each Y value of these points indicate the number of rows of Z coordinate values. In the system, for each element of X, Y and Z coordinates combined in a matrix form, .txt file is written separated by commas. With the writing process, coordinates are transferred to the file. Processing of images and obtaining points are based on the algorithms of edge-hole model. If the processed area is between edge and hole of sample pieces which have edge and hole elements, program uses the surface-area model algorithm.

Surface-Area Model

Surface-area model is an algorithm which processes surface area of front view of pieces and which is used to obtain point cloud from this area. It is completely based on a superficial structure and obtains surface points from the front and back view of the processed piece. In surface area processing, edge detection is not done, and marking line is done randomly between the points marked with pixel detection. The pixels, which are obtained by color analysis done during the processes in the edge-hole model, are used in this model (fig.7).

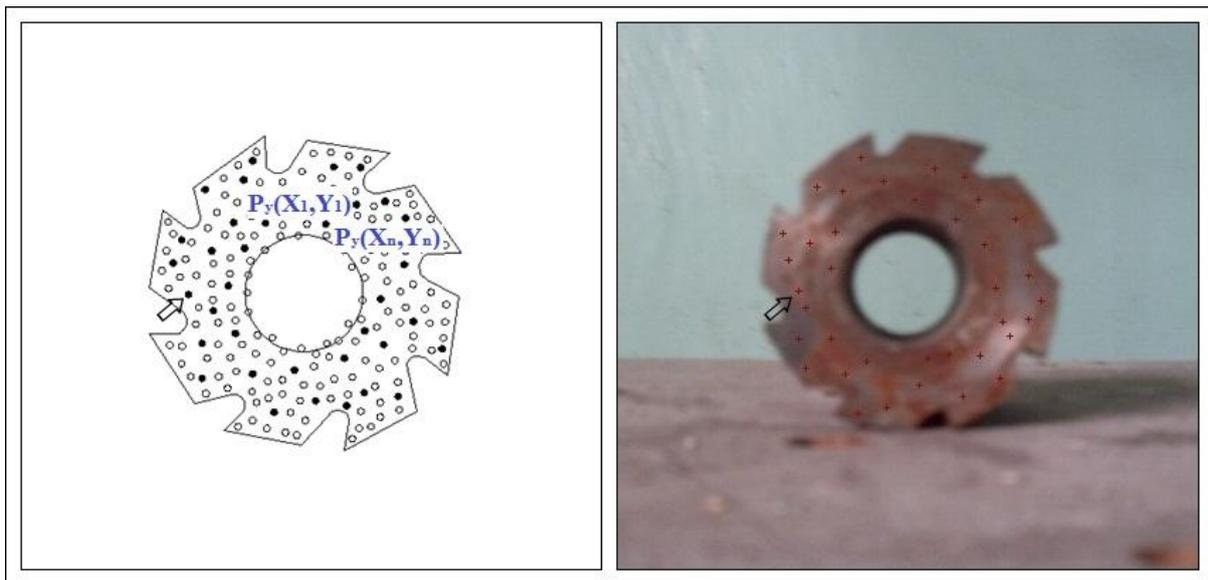


Figure-7: Pixel distribution in the processed image and surface area models

In fig. (7), the marked reference points for surface-area model are shown together with the distribution of pixels. $P_y(X_i, Y_i)$ reference points are shown where filled point are marked as much as the number of $P_y(X_n, Y_n)$ to represent the object the best. The other hollow points are the pixel points detected by color analysis. As in the objects having inner elements as hole, pixel detection is made in the hole. The area, where point detection is made, is the area between inner hole edges (fig.7). For this reason, reference points marked for surface area should be able to indicate and express the analysis area. Reference points identify the area which will be examined. The system acquiring reference points puts these points to the neighboring pixels, which are detected by color analysis the closest to these reference points. In this way, the pixel points acquired from reference points, and color analysis are determined in the following matrix.

$$P_y(X, Y) = \begin{bmatrix} P_y(X_1, Y_1) \\ P_y(X_2, Y_2) \\ \dots \\ P_y(X_n, Y_n) \end{bmatrix}$$

In Eq.12, reference points marked on the surface area and the pixel points which are taken as neighbor to these reference points are shown. Neighbor pixels taken with these reference points can be included within the borders of hole in the pieces where internal elements such as hole are available (fig.7). The point cloud taken in such cases, cannot define the object exactly because of the possible pixel overflows and hence modeling is corrupted. Therefore, structural element is applied to the obtained pixel points as in edge-hole model. With this applied structural element, unwanted pixels are eliminated (fig.8).

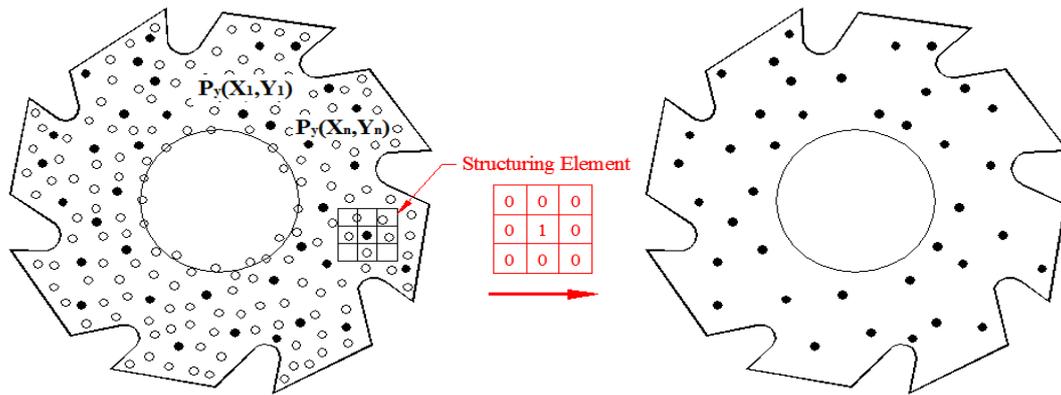


Figure- 8: Applying structural element to the obtained surface pixels

In fig. (8), reference points and pixel distribution before and after the structural element applied to the neighbor pixel of reference points are shown. Panning structural element on the curved and different formed objects is difficult. Therefore, cleaning unnecessary surface pixels from these different edge structured objects is performed by panning marked reference points of structural element. This panning process is done for each reference point. In this way, unnecessary pixels are removed from the image and pixel points from the first stage are simplified. Coordinate values are obtained by taking each X and Y points from the simplified pixel point matrices ($X_{y(coor)}$, $Y_{y(coor)}$).

$$X_{y(coor)} = \begin{bmatrix} X_{y1} \\ X_{y2} \\ \dots \\ X_{yn} \end{bmatrix}, \quad Y_{y(coor)} = \begin{bmatrix} Y_{y1} \\ Y_{y2} \\ \dots \\ Y_{yn} \end{bmatrix}$$

X and Y coordinate values of pixel points obtained from surface-area model are shown in the matrix. With the removed pixels, X and Y values of simplified pixel point matrix are defined as 'n' in the different matrixes. Surface-area model uses the two reference points taken from side view for Z coordinate value (fig. 9).

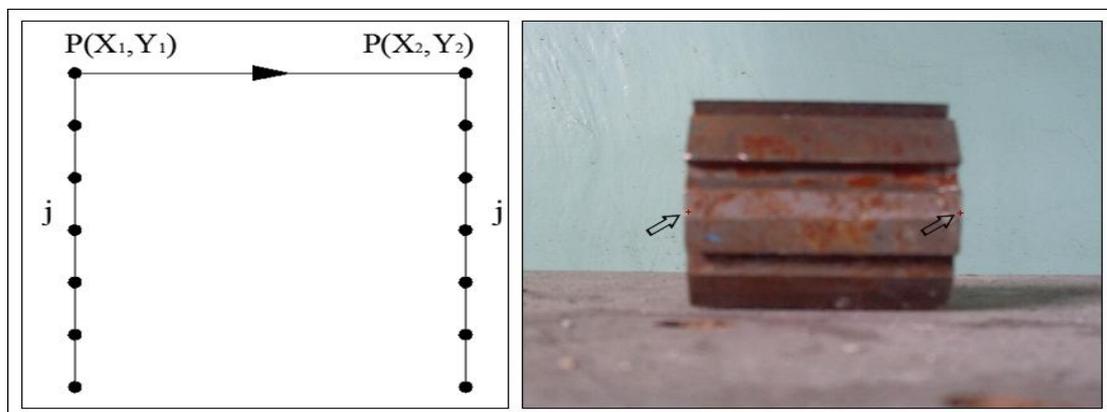


Figure: 9: Pixel detection in side view for surface-area model

In fig. (9), pixel detection for side view in surface-area model is shown after the reference point marked on the image. For side view, two types of Z value are determined. The first Z value is as X_1 and this X_1 value is also coordinate value of front view of the piece and is accepted as 0. Regardless of the pixels between two reference points marked for side view in the second Z value, the distance from X_1 to X_2 are taken. In this way, the pixels belonging to surface area detected in the front view of the piece are determined for the back side at the same time. These obtained Z values are added to X and Y values obtained from the front view, so point matrix is obtained as X_y, Y_y, Z_y matrix form.

$$X_y, Y_y, Z_y = \begin{bmatrix} X_{y(n,1)} & Y_{(n,2)} & a \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} \Rightarrow a = X_2 - X_1$$

In the Eq. 14, coordinate values, which are obtained from surface areas, are shown. 'a' is the difference between two X values marked in the side view. The first Z value is as 0, and the second Z value is as 'a'. The first Z is added to each X and Y coordinate values and obtained point number is written down one under the other in the matrix. Again, the same X and Y values are expressed with the second Z value, and it is added under the first written values, and matrix is maintained. X and Y coordinate values obtained from the front view; these binary values (X_2-X_1) are used and these values are taken in a matrix by combining one under the other and so X,Y and Z coordinate values are obtained. These points obtained by the surface-area model are combined with the points acquired from the edge-hole model and point cloud of processed object is achieved (fig.10).

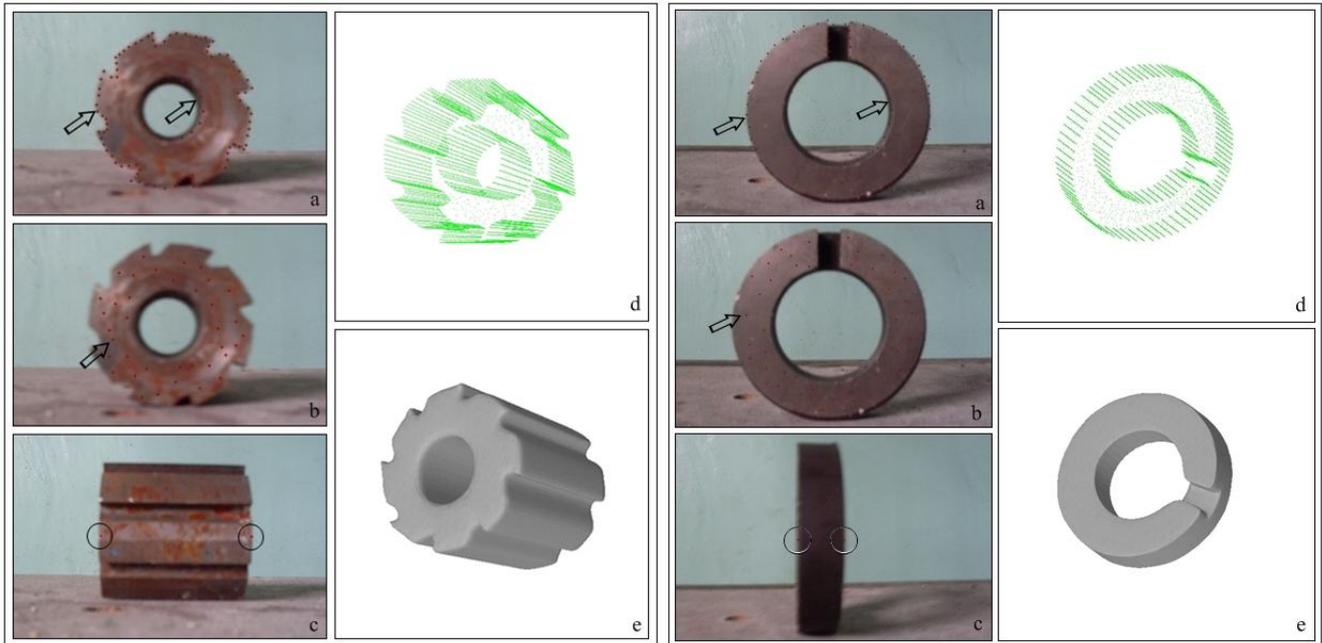


Figure 10: Obtaining point cloud and solid model; a. Front view edge-hole model, b. Front view surface-area model, c. General reference points for side view, d. Point cloud, e. Solid model

In fig. (10), reference points marked by the user on the image are specified, after processing the image acquired point clouds are shown with their solid models. Solid models are generated through a computer aided design program by using obtained point clouds.

IV. RESULTS AND DISCUSSION

In the developed image processing system, images taken with a digital camera in an ordinary environment without the need for calibration are evaluated easily. With the algorithms developed for evaluating the image, point clouds of many pieces, which are expressed in two appearances from simple to complex, are can be obtained. The number of marked reference points may vary depending on the marked images. This case is proportional with the number of edge and hole of the processed target object. If the number of marked reference point is too much, this shortens the time process of the system. After reference points are marked, the elapsed time to obtain 3D point clouds is not more than 4 seconds.

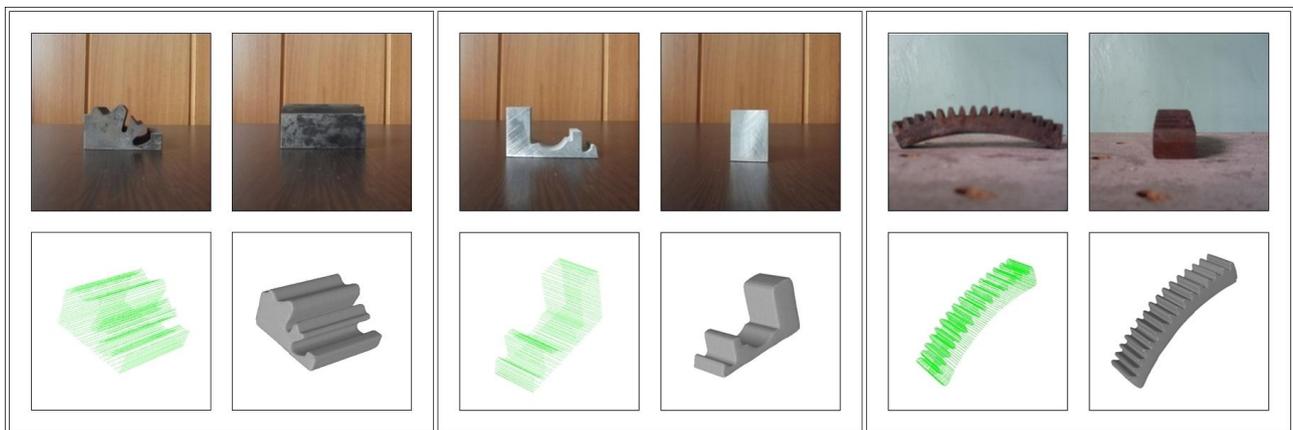


Figure 11: Point clouds obtained from the images not containing element of hole and their solid models

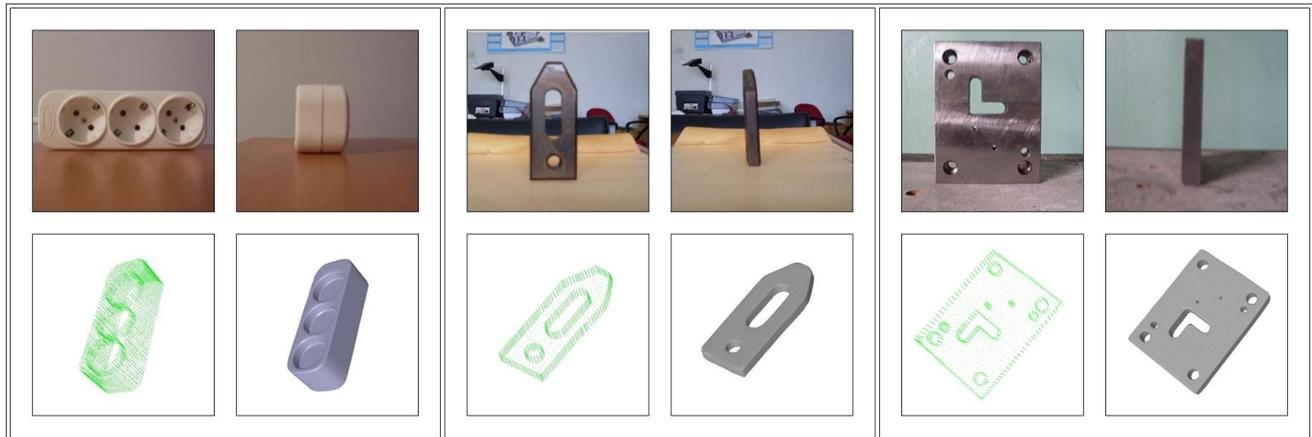


Figure 12: Point clouds obtained from the images containing element of hole and their solid model

The processed sample images and obtained point clouds are shown with their solid models in fig. (11) and fig. (12). The elapsed time to remove the point clouds is between 2-4 seconds. Sample images were taken in natural light environment and were taken by placing on a flat surface. The system can detect the elements of the area which will be processed as on and off, and it evaluates the images by using reference points, color analysis and developed algorithms. In many studies where image processing and interpreting digital images are available, special shooting environment and calibration for the camera are used [22]. Camera calibrations are made by using such techniques as grid projection, markers placed on the target objects [23]. At the same time, markers are one of the helper tools used in non-contact systems which is a Reverse Engineering application. In other studies, object libraries are formed for the target object so that the objects in the images can be identified. In evaluating and comparing similar form of objects, object libraries are frequently used [4]. In addition, edge detection algorithms are used to determine the target object as 2D [24]. In this study, to obtain 3D point cloud of objects, such engineering environment as calibration, taking pose, monitoring systems and positional markers are not required.

V. CONCLUSION AND FURTHER WORK

In developed program, effective use of elapsed time, which is for design and manufacturing, reduces costs in an effective way. One of the main reasons for the use of reverse engineering applications, it shortens the design process. The studies intended to shorten the time spent on the design stages of products will affect manufacturing process in an efficient way. The advantages provided by the developed system are as follows:

- In the program, point clouds can be achieved from the simple, and complex structured pieces
- The pieces, which will be designed in the computer environment or manufacturing operation, will be applied, are transmitted more quickly into the design environment with developed system. If desired computer-aided static, dynamic analysis procedures and manufacturing processes can be performed. In addition, obtained model can be loaded to rapid prototyping machines and prototype of the object can be obtained.
- Images with different resolution are used in the program and obtained point clouds express the target object in the image fully.
- In developed program, with the use of reference point method, image interpreting area is limited and only by processing the target area, undesirable point clouds, which are formed by the elements of background of the image, are removed. This reduces the processing time to 2-4 seconds, and at the same time, it enables the images taken to be processed in the program faster.

The new modules can be added to the program and with the developed new algorithms, processing the images expressed in more appearances and having more complex geometry can be achieved. The image processing technique, in which developed algorithms and color analysis are used, will be beneficial for the future studies in this field. The images processed in the program do not have resolution limit, so it will allow to processing the high-resolution images in parallel with technology. Next study target is to obtain 3D point clouds from completely cylindrical and curvilinear structured and more complex object images by adding a new module to the developed software.

References

- [1] V. Raja, Introduction to reverse engineering, in V. Raja and K.J. Fernandes (Eds.), Reverse engineering an industrial perspective, (London: Springer, 2008) 1-9.
- [2] S. Motavalli and R. Shamsaasef, Object-oriented modeling of a feature based reverse engineering system. *Int J Comp Integ Manufac*, 9(5), 1996, 354-368.
- [3] R. Clouard, A. Renouf and M. Revenu, Human-computer interaction for the generation of image processing applications, *Int J Human-Comp Studies*, 69, 2011, 201-219.
- [4] L. Hua and W. Weiyu, A new approach to image-based realistic architecture modeling with featured solid library. *Auto in Const* 13, 2004, 555-564.
- [5] G. Zhou, Primitive recognition using aspect-interpretation model matching in both CAD and LP based measurement systems, *ISPRS J of Photog and Rem Sensing* 52, 1997, 74-84.

- [6] D. Frere, J. Vandekerckhove, T. Moons and L. Van Gool, Automatic modeling and 3D reconstruction of urban buildings from aerial imagery, IEEE Int Geoscience and Rem Sensing Symp Proc, Seattle, 1998, 2593-2596.
- [7] A. Huertas, R. Nevita and D. Landgrebe, Use of hyperspectral data with intensity images for automatic building modeling, Proc of the Sec Int Conf on Inf Fusion, Sunnyvale, 1999, 680-687.
- [8] G. Bleser, Y. Pastarmov and D. Stricker, Real-time 3D camera tracking for industrial augmented reality applications, J of Winter Sch of Compt Graphics, 2005, 47-54.
- [9] R. Azuma, A survey of augmented reality, Computer Graphics SIGGRAPH Proc, 1995, 1-38.
- [10] U. Bockholt, A. Bisler, M. Becker, W.K. Müller-Wittig and G. Voss, Augmented reality for enhancement of endoscopic interventions, Proc IEEE Virtual Reality Conf , Los Angeles, 2003, 97-101.
- [11] S. Wesarg, E. Firle, B. Schwald, H. Seibert, P. Zogal and S. Roeddiger, (2004) Accuracy of needle implantation in Brachtherapy using a medical AR system-a phantom study, SPIE Medical Imaging Symp, USA, 2004, 341-352.
- [12] L. Song and D. Wang, A novel grating matching method for 3D reconstruction, NDT & E Int, 39, 2006, 282-288.
- [13] Z. Xie , J. Wang and Q. Zhang , Complete 3D measurement in reverse engineering using a multi-probe system, Int J of Mach Tools Manuf , 45, 2005, 1474-1486.
- [14] L. Song , X. Qu, K. Xu and L. Lv, (2005) Novel SFS-NDT in the field of defect detection, NDT&E Int., 2005, 38:5 381-386.
- [15] L. Song , X. Qu, Y. Yang, C. Yong and S. Ye, (2005) Application of structured lighting sensor for online measurement, Opt and Laser in Eng, 43,10, 2005, 1118-1126.
- [16] A.P. Witkin, Scale-space filtering. Proc of 8th Int Joint Conf Art Intel Karlsruhe, Germany, 1983, 1019-1022.
- [17] T. Lindeberg, Scale-space for discrete signals, IEEE Trans Patt Recog Mach Intel 12, 1990, 234-254.
- [18] B.R. Barbero and E.S. Ureta, Comparative study of different digitization techniques and their accuracy, Computer Aided Design, 43, 2011, 188-206.
- [19] F. Bosche, Automated recognition of 3D CAD model objects in laser scans and calculation, Adv. Eng. Informatics, 24, 2010, 107-118.
- [20] W-B. Yang, M-B. Chen and Y-N Yen, An application of digital point cloud to historic architecture in digital archives, Advn in Eng Soft, 42, 2011, 690-699.
- [21] W.K. Pratt, Digital image processing, A Willey-InterScience, New York, 1991, 490-530.
- [22] K. Lee, K. Wong , S.Y. Fung, (2001) 3D face modeling from perspective-views and contour based generic model, Real-Time I, 7, 2001, 173-182.
- [23] L. Song and D. Wang, A novel grating matching method for 3D reconstruction. NDT & E Int., 39, 2006, 282-288.
- [24] D. Tubic, P. Hébert , D. Laurendeau , 3D surface modeling from curves, Img and Vis Comp 22, 2004, 719-734.