www.s2is.org
International Journal on Smart Sensing and Intelligent Systems
ISSN 1178-5608

SURFACE FOLLOWING WITH AN RGB-D VISION-GUIDED ROBOTIC SYSTEM FOR AUTOMATED AND RAPID VEHICLE INSPECTION

Danial Nakhaeinia, Pierre Payeur, Alberto Chávez-Aragón,
Ana-Maria Cretu, Robert Laganière, and Rizwan Macknojia
School of Electrical Engineering and Computer Science,
University of Ottawa, 800 King Edward, Ottawa, Ontario, Canada
Emails: [dnakhaei, ppayeur, achavez, acretu, laganier, rmack102]@uottawa.ca

Submitted: Feb. 1, 2016 ***Accepted: Apr. 12, 2016*** ***Published: Jun. 1, 2016***

Abstract- This paper presents the design and integration of a vision-guided robotic system for automated and rapid vehicle inspection. The main goal of this work is to scan and explore regions of interest over an automotive vehicle while a manipulator's end effector operates in close proximity of the vehicle and safely accommodates its curves and inherent surface obstacles, such as outside mirrors or door handles, in order to perform a series of close inspection tasks. The project is motivated by applications in automated vehicle inspection, cleaning, and security screening. In order to efficiently navigate the robotic manipulator along the vehicle's surface within regions of interest that are selectively identified, an efficient and accurate integration of information from multiple RGB-D sensors and robotic components is proposed. The main components of the proposed approach include: automated vehicle category recognition from visual information; RGB-D sensors calibration; extraction of specific areas to inspect over the vehicle body, and path planning from an efficiently

reconstructed 3D surface mesh to move the robotic arm along and in close proximity of the vehicle. The proposed multi-stage system developed merges all components to achieve rapid 3D profiling over a complex surface in order to fully automate the process of surface following for vehicles of various types and shapes. To validate the feasibility and effectiveness of the proposed method experiments are carried out with a 7-DOF manipulator navigating over automotive body panels.

Index terms: Security robotics, sensor based navigation, RGB-D imaging, curvature estimation, path planning, surface following.

I. INTRODUCTION

Detecting dangerous or prohibited substances hidden in a vehicle without direct human intervention is a critical asset to ensure the security of populations and properties worldwide. Some public and private institutions such as government buildings, research centers, military bases, airports, and critical infrastructures require rigorous vehicle or individual screening systems at their periphery. While very efficient technologies exist to detect the presence of minute amounts of dangerous material particles, the process of collecting such particles safely and efficiently in an automated manner remains a challenge, especially over large surfaces such as a vehicle. This paper proposes an efficient and automated integrated vision-guided robotic system driven by multi-modal sensing to fully automate the detection of potentially compromising particles over an automotive vehicle in the context of security screening in restricted access areas. While one of the major objectives is to make the process safer for the operators, the proposed solution also aims at providing authorities with pre-event screening mechanisms for prohibited substances while remaining versatile, easy to use, and permitting automated screening of large vehicles in time critical applications.

In order to accomplish such complex operations, the robotic system must be able to sense the environment and analyse the information accurately, but still rapidly. In the literature, a large range of sensors are frequently employed to support such complex tasks. Unfortunately, many types of sensors are not capable to capture color and depth images simultaneously, or are slow and require a considerable amount of time to perform range acquisition over a surface of the size of an automotive vehicle. To overcome these limitations, several attempts have been made to capitalize on novel consumer-grade RGB-D cameras such as the Microsoft Kinect sensor. This sensor is able to generate high density depth maps and corresponding color images in a fraction of a second. In the context of vehicle security screening considered here, which imposes tight time constraints, this technology can be beneficial, in spite of its inherent operational limits. Nevertheless, typical vehicles size largely exceeds the field of view of any RGB-D sensor. Therefore a multi-view vision system is required to rapidly acquire and reconstruct a full scale model of an entire vehicle undergoing inspection. The solution proposed in this work involves a network of rapid and relatively low resolution RGB-D sensors. The raw data being rapidly captured by a calibrated network of Kinect sensors makes it possible to constrain the screening procedure to only a few minutes, while still allowing for the process to determine the category of the vehicles (e.g. car, van, SUV) and provide the required 3D information for the robotic surface following to proceed with sufficient accuracy. The information collected includes the position and orientation of the surfaces, their shape and normal direction, and also the location of obstacles that must be taken into account for path planning and surface following. The compromise made to achieve sufficient speed for the acquisition and screening procedure was to adopt relatively low resolution sensors to acquire the depth maps. This has an impact that must be dealt with at the path planning stage. The proposed approach is designed and experimentally validated in the specific context of automotive vehicle surface shape following but is applicable to a wide diversity of shapes and object sizes. Within the context of development considered, the solution automatically adjusts the screening procedure for vehicles of various types, brands and sizes.

The main contributions of the work are in the development of a dexterous manipulation framework and algorithms for exploration, scanning and shape estimation of miscellaneous object surfaces endowed with a robust motion planning and control approach coupled with RGB-D vision sensing. This paper closely examines the integration considerations of the vision stages and the robotic components to perform the security screening process efficiently and in a coordinated way.

The paper is organized as follows: Section 2 describes related work. Section 3 provides a formal problem formulation and set of requirements. Section 4 details the system design. Experimental results are presented in section 5, and a final discussion concludes the paper in Section 6.

II. RELATED WORKS

To achieve complex manipulation with everyday life objects, robotic systems must become more aware of their environment and, as such, earn extra flexibility to fully capitalize on their degrees of freedom. For that matter, it is essential to develop innovative integrated sensing and control methods to provide next generation robots with increased versatility that will make them perform closer to what human beings can achieve. Computer vision and embedded force and torque sensing have been widely explored to assist robotic manipulation [1]. However, limitations remain with these sensing solutions, often resulting from the lack of adequate sensing capabilities to ensure robust interaction control when the robot is in close proximity or in contact with an object. The problem becomes even more prominent as modern robotics calls for operation in a priori unknown workspaces, which involves a full range of sensing, planning and fine control steps that must be performed by proper planning and control algorithms.

For cases involving surface shape following, two types of sensors are generally used: touch (force/torque and tactile) and vision (2D/3D cameras). Force/torque sensors are critical components to extend the capability of robot manipulators. With the help of these sensors a robot can deal flexibly with uncertainties in its environment and execute complicated tasks. Force/torque feedback is often used to compensate for positioning inaccuracies in simultaneous position and force control of robot manipulators. In [2], the problem of force and position control of a robot manipulator performing compliant tasks is addressed which describes a 3D surface tracking controller based on the Smith predictor design. Yin et al. [3] propose a methodology of active tracking of unknown surface using force sensing and control technique. They present a strategy for force sensing and control on the basis of information fusion of force and position, fuzzy hierarchical coordination and neural control. In similar works, Papageorgiou et al. [4] propose a methodology to drive the end-effector of a non-redundant manipulator in close proximity of a surface while avoiding obstacles. Once the end-effector is close to the surface, a second controller takes over to stabilize the end-effector at a predefined distance to the surface. Later on, in [5], force control is added for tasks that involve dealing with contact on surfaces.

In manipulative tasks, tactile information not only allows the assessment of object properties, but also guides manipulation based on contact information. It is used as a control parameter to obtain contact point estimation, surface normal, curvature measurement, and slip detection through measurement of normal static forces [6-8]. In [9] the problem of autonomously estimating position and orientation of an object from tactile data is addressed for 6-DOF localization using a Bayesian approach. Wang and Li [10] present an approach to control a manipulator tracking a surface where the tactile sensing is integrated with force-torque information as the feedback. Hybrid impedance control is applied to follow both the position and force trajectories. Later, they extended the work to control the end-effector of a redundant manipulator in [11]. However, the force/torque and tactile sensors are only applicable in the context of surface following when there is physical interaction between the robot and the objects located in its workspace. An initial contact point between the tool and the object must be prescribed and the force controlled contact must be maintained. Therefore, due to the limited bandwidth of the contact or force sensor, the execution speed of the task is limited to prevent loss of contact and information. Furthermore, these methods are applicable only where planned trajectories are known beforehand [12].

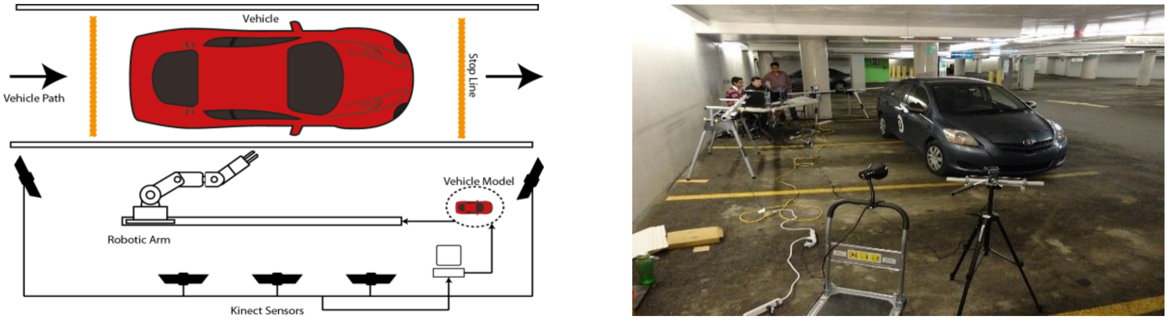
A fundamental advantage of using vision-based control is that no contact with the object is required as it allows non-contact measurement of the environment. Vision sensors provide global or local information to determine the robot and objects relative locations in the environment. This information is used by the robot controller to reach the object or target while avoiding undesired obstacles. Most of the proposed approaches on vision control for robot manipulators have involved free motion control. These approaches demonstrate a good performance in dealing with planar surface following but are not very successful over 3D structures because the depth of the object is also required for precise path planning. Therefore, hybrid vision/force controllers [13-15] have been proposed to deal with contact tasks [16] which impose delays due to the limited contact points.

Alternatively, several solutions for vision-guided robotic inspection make use of high-cost 3D profiling cameras, scanners, sonars, or combinations of them, which often results in lengthy acquisition and slow processing of massive amounts of information. However, the ever growing popularity and adoption of RGB-D sensors, especially the Microsoft Kinect sensor, recently motivated its introduction in the development of vision-guided robotic systems. Numerous examples of application for the Kinect technology recently appeared in the literature [17-19]. For the development of the system presented here, the extreme acquisition speed of the Kinect technology and its low cost have been major selection criteria for this sensor to be used, given that rapidly acquiring color and 3D data over large volumes is instrumental.

The robot path planning methods typically used for surface following can be divided into two categories: learning-based and automatic. In learning-based path planning [20-23] methods, the robot is usually trained by the operator, or from offline generated paths, to scan a sample surface, and the same quality and shape is expected for the subsequent objects. These methods rely on the operator's experience and knowledge of the facilities and equipment, on their capabilities, as well as on the processes and tools. In this case, the robot is trained using different methods which involve extensive tests and iterations of the teaching process until the best results are achieved. This process is complex and time consuming while the results vary based on the operator's skill and the learning process. To overcome these shortcomings, it is desirable to use an automated path planning process. Automating the planning process reduces human labor dramatically, and keeps human operators from being exposed to harmful working environments. Hence, it is essential to replace the manual path planning methods by automated ones. This challenging research topic has been receiving more and more attention from academia and industry. As part of these initiatives, this paper introduces a fully automated vision-guided global path planning method that is adapted to explore and scan regions of interest over a vehicle body using a rapidly constructed 3D model of the vehicle. As such, this paper represents a technically extended version of our previous paper [24] on the complete development and integration of the sensing stage and the path generation process to handle more complex surface shapes.

III. PROBLEM FORMULATION AND REQUIREMENTS

The aim of this work is to rapidly collect data over the surface of a vehicle in order to efficiently navigate a robotic manipulator toward regions of interest that are selectively identified as being critical for security screening. A set of RGB-D sensors are configured to interact as a collaborative network of imagers to achieve full coverage and rapid 3D profiling of automotive vehicles within a short response time [25]. Fig. 1a shows the layout of the proposed automated vehicle screening station and Fig. 1b presents an instance of the physical implementation of the RGB-D sensing stage while conducting experimental evaluation of the vision stage. The vehicle is stopped in front of a set of Kinect RGB-D sensors while the manipulator can be moved on a linear track placed aside the vehicle to extend its reach. The manipulator is initially located at the extremity of its workspace to avoid creating any occlusion during the RGB-D data acquisition phase. The network of Kinect sensors distributed around the vehicle is properly calibrated to collect the color and depth information over a 180 degrees view of the vehicle (one side) by five calibrated Kinect sensors within only a few seconds. A duplication of the same setup can be deployed on the other side of the vehicle to acquire its complete surface. The color and depth information is then processed to construct a textured 3D model of the vehicle. The specifications initially set for the task were that the entire procedure of acquisition, modeling, and robotic interaction is completed within 2 to 3 minutes, and that the accuracy achieved on the 3D reconstructed model is sufficient to navigate the manipulator while following the surface of the vehicle's bodywork.



(a) (b)

Figure 1. a) Design of the vehicle screening station, and b) physical implementation of the RGB-D scanning station.

IV. SYSTEM DESIGN

The assembly of the main components of the proposed RGB-D vision-guided robotic platform for vehicle inspection is presented in Fig. 2. The main components of the system are: vehicle category recognition functionalities for automated and efficient recognition and classification of different vehicles; a registration module using color and depth data from Kinect sensors to estimate the calibration parameters of the vision stage; a visual detector of vehicle parts (VDVP) to detect specific areas that will be inspected over the vehicle; an efficient surface mesh generator for 3D modelling; and a global path planning algorithm that operates from a 3D surface mesh to achieve precise robot control. These components are detailed in this section.

a. Vehicle category recognition

Given that an automated security screening process requires the verification of various areas over a vehicle (such as door handles, seem of doors, trunk area), the procedure must be adapted to the specific shape of a vehicle entering the inspection station in order to perform reliably while still meeting tight time constraints. With the large number of vehicles circulating over our roads and the diversity of their categories and size, it appears essential to first recognize to what category a vehicle belongs (e.g. sedan, SUV, minivan, small truck) before collecting any further data. For this purpose, a vehicle classification system from color images collected from different views that has been recently proposed [26] was adapted to support the security screening application. The computational approach consists in the use of human visual attention mechanismsto detect discriminative salient features [27], and a set of support vector machines (SVM) classifiers, complemented by a width and height-based discriminating heuristic, to achieve fast automated classification. As shown in Fig. 3, five images are collected by the 5 color cameras of the Kinect sensors distributed around the vehicle (Fig. 1) for multi-view classification. To identify a predetermined number of features for each view, saliency maps (SM) are obtained using Itti et al. [28] visual attention model. The number of salient features required for classification of the images in a dataset is defined using the saliency threshold, S_T (Eq. 1). To build a limited m-feature saliency map, SM_l (Eq. 2), all the selected salient features in SM are replaced with 1s and the rest of the points with 0 for each view of a vehicle:

$$s_T = \sum_{i=1}^{m} s_i \,/\, \sum_{j=1}^{n} s_j \tag{1}$$

$$SM_l(x,y) = \begin{cases} 1 & if\ SM(x,y) \in S_m \\ 0 & otherwise \end{cases} \tag{2}$$

where $s \in S$, $S = \{s_k |\ s_k = SM\ (x_k\ ,\ y_k),\ s_k > s_{k+1},\ k=1\ldots n\}$, m is the number of salient pixels to be considered, and n is the total number of pixels in the saliency map, SM, with $m < n$. The number of salient features, m, is defined such that at least 99.8% of the images in a dataset reach a saliency threshold, s_T, of at least 0.5. S is a list of all the pixels in SM from the most salient to the least salient.

To perform a binary classification, the set of features extracted from each image is transformed into a vector and 5 SVM classifiers are trained for each view. The classifier output is 1 if it recognizes the vehicle in each view and 0 otherwise. Finally, the results of all 5 classifiers are composed from each view which is called a confidence measure. The confidence value changes from 0 to 5. When none of the classifiers identifies the vehicle in a category, the confidence is 0, and when all the classifiers recognize the vehicle, it is 5. The vehicle belongs to the category that provides the highest confidence measure. If 2 or more categories provide the same confidence measure, two additional SVM classifiers are trained to identify the vehicle category based on the width and height. The saliency map obtained above is used to estimate the width and height of the vehicle. The classifier receives a vector composed of the width and height of a vehicle computed based on the feature set in each image to map the vehicle to a corresponding category.

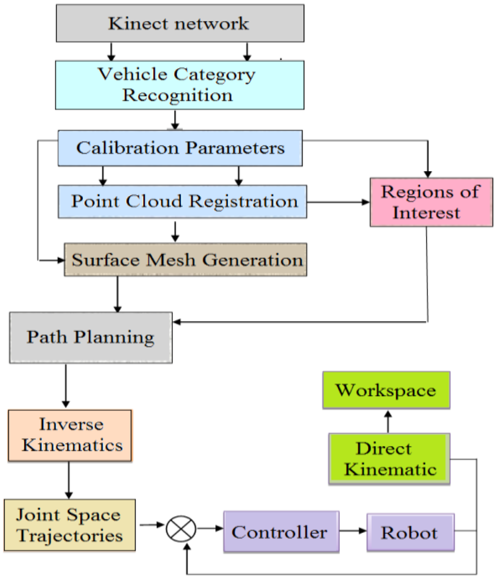
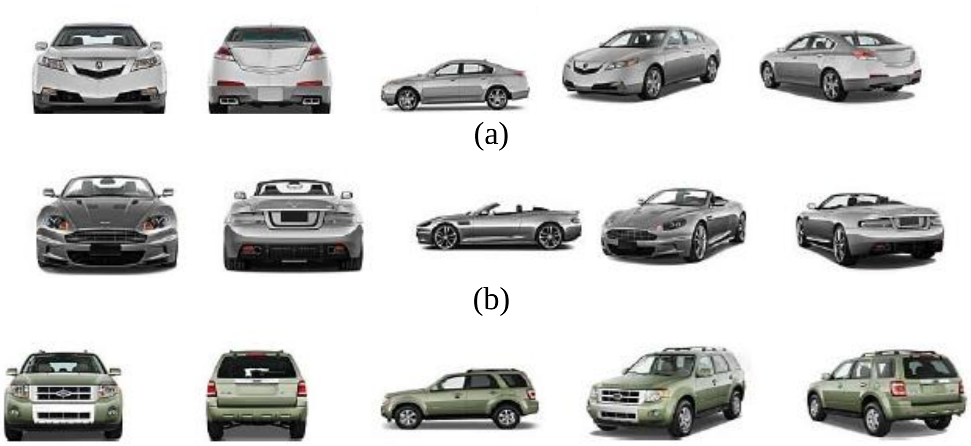


Figure 2. Block diagram of the vision-guided surface following robotic system.



(a)

(b)

(c)

Figure 3. Multi-view vehicle classification: examples of vehicle categories: a) sedan, b) sports car, c) SUV.

Our early work validated the method on vehicles of three different categories: sedan, sport car, and SUV. The overall recognition rates reached 96% where only salient features were used, and over 99% correct classification when using the additional estimations on the width and height of the vehicles [26]. As such, the approach demonstrated superior performance to alternative solutions from the literature.

b. Calibration & data registration

In order to coordinate the movement of the manipulator using rapidly acquired 3D data by the network of Kinect sensors, the corresponding textured 3D point clouds must be accurately registered with respect to the base of the robot. For this purpose, a calibration methodology was proposed in [25] to estimate the internal and external calibration parameters over a network of Kinect sensors to achieve accurate alignment between the respective point clouds and textured images acquired by each Kinect sensor that are distributed in a collaborative network of imagers to provide coverage over the large volume occupied by the vehicle.

Extrinsic parameters estimation for built-in Kinect sensors (color and depth) within each Kinect unit is determined by stereo calibration. The position between both cameras is defined by (3):

$$H = H_{RGB} H_{IR}^{-1} \tag{3}$$

H_{IR} is the homogenous transformation matrix from the depth (IR) camera to a checkerboard target used for calibration, and H_{RGB} is the homogenous transformation from the color (RGB) camera to the checkerboard target. The internal extrinsic calibration parameters allow to accurately relate the color to depth data collected by a given Kinect device. However, the Kinect sensor does not directly provide the registered color and depth images. Therefore, the 3D coordinates corresponding to each point in the depth image are computed as follows:

$$\begin{aligned} X_{IR} &= (x - O_{x\_IR}) du(x,y) / f_{x\_IR} \\ Y_{IR} &= (y - O_{y\_IR}) du(x,y) / f_{y\_IR} \\ Z_{IR} &= du(x,y) \end{aligned} \tag{4}$$

where (X_{IR}, Y_{IR}, Z_{IR}) are the 3D point coordinates of pixel (x, y) in the depth image with respect to the IR camera reference frame, $(x,\ y)$ is the pixel location in the depth map, $(f_{x\_IR},\ f_{y\_IR})$ is the focal length of the IR camera, (O_{x_IR}, O_{y_IR}) is the optical center of the IR camera, and $du(x,y)$ is the depth of a pixel in the depth image. Next, the color is assigned from the RGB image to each 3D point $P_{IR}(X_{IR}, Y_{IR}, Z_{IR})$ as follows:

$$\begin{aligned} P_{RGB}\ (X_{RGB}, Y_{RGB}, Z_{RGB}) &= R \cdot P_{IR} + T \\ x &= (X_{RGB} . f_{x\_RGB} / Z_{RGB}) + O_{x\_RGB} \\ y &= (Y_{RGB} . f_{y\_RGB} / Z_{RGB}) + O_{y\_RGB} \end{aligned} \tag{5}$$

where P_{RGB} is the 3D point with respect to the color camera reference frame, R and T are the rotation and translation parameters from the color camera to the depth camera obtained from the calibration procedure, (f_{x_RGB}, f_{y_RGB}) is the focal length of the color camera, (O_{x_RGB}, O_{y_RGB}) is the optical center of the color camera, and (x, y) is the pixel location of color information in the color image.

The last set of parameters estimated in the calibration process is the extrinsic one, which is the relative position and orientation between every pair of Kinect sensors. This way a complete calibration of all cameras in the network of Kinect sensors is achieved [25]. As shown in Fig. 4, the center Kinect, K_1, is set as a base of reference for the setup. The relative calibration is then calculated between (K_1, K_0), (K_1, K_2), (K_2, K_3) and (K_0, K_4).

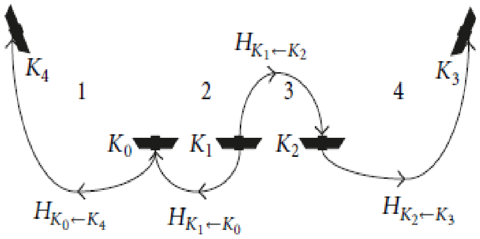


Figure 4. Complete calibration of all sensors in the network of Kinect sensors.

The raw Kinect sensor information is stored in a textured 3D point cloud form. The vision stage is also calibrated with the robot's base reference frame during the setup procedure. Therefore, all the points imaged on the vehicle are defined as Cartesian coordinates with respect to the robot's base reference frame, which allows for path planning and guidance of the manipulator from the visual stage. As shown in Fig. 5, the calibration parameters are initially estimated with the method detailed in [25] and then finely tuned with the use of an iterative closest point (ICP) algorithm as part of the calibration process. The calibration procedure is performed offline in order to meet the vehicle screening time requirements before the inspection station is put in operation. As such it does not impact the execution time for the security screening process. The refined calibration parameters are later used to perform actual registration between the piecewise datasets originating from the 5 separate Kinect sensors in order to guide the manipulator from visual information.

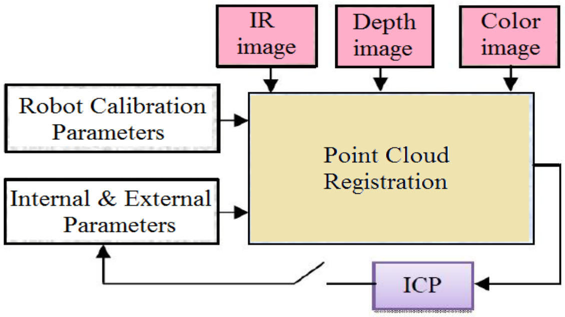


Figure 5. Calibration and data registration module.

c. Regions of interest extraction

Performing a screening operation over the entire surface of the vehicle remains unrealistic within the time constraints imposed by the tasks. Given the nature of the operation, a number of critical areas were selected given the higher probability of finding prohibited substances in those regions that are prone to hands contact with the vehicle. In order to automatically and efficiently locate those targeted areas over a vehicle, a visual detector of vehicle parts (VDVP) has been developed [29], which also takes advantage of the fully calibrated network of Kinect sensors.

The VDVP method determines the location of the vehicle in the scene and subsequently the location of a set of significant vehicle components. It initially relies on the successful detection of the car wheels from the lateral views. The detector uses the Hough transform algorithm and a classifier based on Haar-like features trained for detecting wheels. The approach estimates the shape over selected regions to be reconstructed in order to support the robot navigation and surface following based on the detection of features of interest on vehicle body panels. The VDVP receives a color image of a lateral view of the vehicle as an input and automatically and efficiently determines the location of up to 14 vehicle characteristic parts that are relevant for security screening (Fig. 6). Once the location of the regions of interest is known, the approach uses inputs from the depth sensors of the Kinect devices to acquire and reconstruct the shape of the vehicle's surface within the regions of interest only, thus speeding up the 3D imaging process.

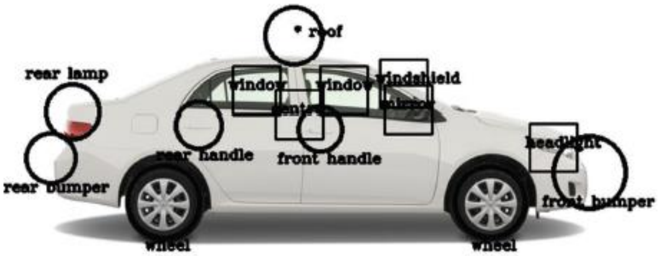


Figure 6. Regions of interest detection over a 4-door sedan car.

The specific areas of interest over a large object such as a vehicle are re-organized in order to speed up the modeling process and to facilitate the guidance of the robot arm. By focusing the operation only over selected areas of importance for the screening procedure the amount of time spent on scanning of the vehicle surface is significantly reduced.

d. Surface mesh generation

The point clouds collected by the set of Kinect sensors are first aligned using the external calibration parameters previously calculated. Then, using the VDVP method, the points contained within the specific areas of interest are segmented from the whole point cloud to make the representation of the surface shape only over those regions of interest. In this stage it is desired to automatically reconstruct the 3D shape over selected regions using the depth information from the global RGB-D model. Fig. 7 illustrates the procedure for 3D reconstruction of regions of interest over the vehicle lateral panels. For this purpose, the locations of the detected parts of interest in color images are mapped unto the 3D point cloud. A surface mesh is created from the registered point clouds using an accelerated meshing technique [29] that takes advantage of the structured organization of depth readings generated by Kinect sensors. A set of local 3D colored models that represent the shape and visual appearance of the surface within each region of interest is obtained.

Rather than using the classical Delaunay triangulation technique, which tends to be computationally expensive, here the triangulation mesh is computed more efficiently by taking advantage of the structured information provided by Kinect sensors. In this case, an individual triangle mesh is built over the corresponding groups of points that belong to each region of interest.

The models are stored in a PLY format file which contains a collection of vertices and faces which are called the vertex list and the face list. The vertex list is a list of (x, y, z) triplets for vertices with respect to the robot base frame. The face list contains the number of vertices that form the face (triangle), followed by the vertex indices that compose each face. The proposed meshing technique generates a mesh for the entire point cloud from one Kinect in about 0.1 second.

A 3D mesh of a mock-up car door panel is shown in Fig. 8, after the Quadric Clustering decimation algorithm [30] is applied over the resulting triangle mesh to reduce the number of triangles for better visualization.

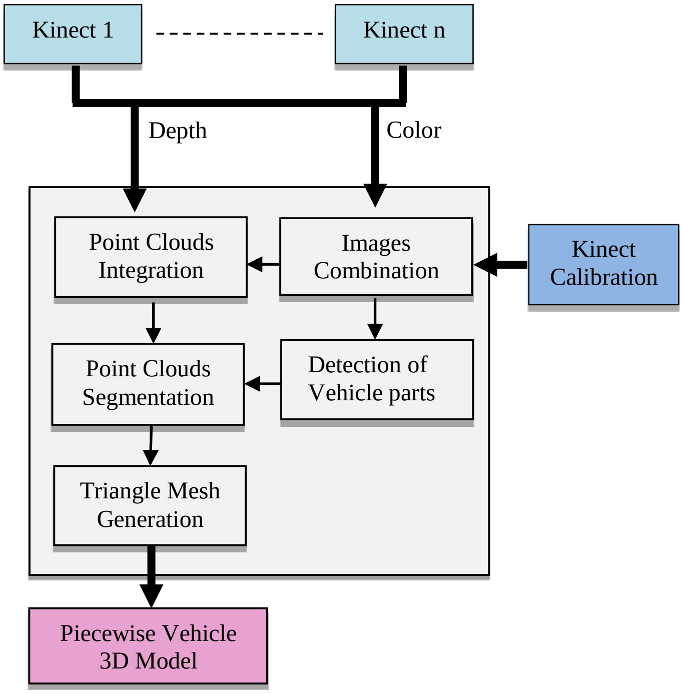


Figure 7. 3D reconstruction of selected regions over the vehicle.

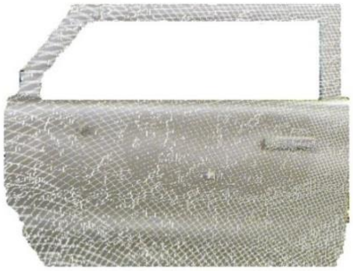


Figure 8. Triangle mesh of a car door model.

e. Path planning for surface following

The piecewise 3D models generated in the previous step are used as inputs for a robotic path planner to allow the manipulator's end effector to efficiently scan a given region of interest in close proximity of a vehicle while performing some inspection tasks. For that purpose, the 3D model of the region is first processed to extract the vertex list and the face list to plan a trajectory in Cartesian space for scanning the whole region of interest. Then a triangle mesh of the surface is formed to compute the normal to each vertex that is part of the planned trajectory. Finally the orientation of the surface area that will be scanned is estimated to compute the orientation of the end-effector at each location while meeting the desired point coordinates over the path.

1) 3D data processing

The vertex list and the face list are used jointly to plan a path for the manipulator and simultaneously form the 3D triangle mesh of reference to estimate the local normal and orientation of the surface which are also used to control the orientation of the manipulator. The 2D representation of a 3D point cloud is presented in Fig. 9. The 3D point of $\boldsymbol{i}$ can be accessed from the vertex list $\boldsymbol{VL}$ as follows:

$$vi \leftarrow VL[index] \tag{6}$$

The black dot is a vertex $\boldsymbol{i}$ shared by the triangles: (e, i, n), (n, i, w), (e, i, s) and (s, i, w) which form four faces, where $\boldsymbol{e}$, $\boldsymbol{s}$, $\boldsymbol{w}$ and $\boldsymbol{n}$ are four neighbor points of $\boldsymbol{i}$. The triangles are defined as follows:

$$\begin{aligned} TM1[j] &\leftarrow triangle(v1, v2, v3) \\ TM2[j] &\leftarrow triangle(v1, v4, v5) \end{aligned} \tag{7}$$

Where $v1 = VL(i)$, $v2 = VL(e)$, $v3 = VL(s)$, $v4 = VL(w)$ and $v5 = VL(n)$.

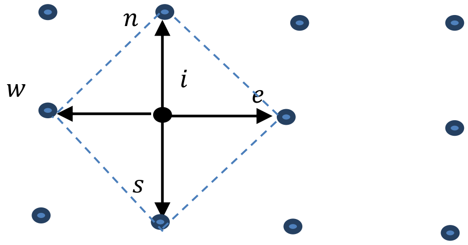


Figure 9. 2D representation of a 3D point cloud.

The triangulation process is performed rapidly using this procedure and later on supports the estimate of local normal and surface curvature that control the orientation of the manipulator.

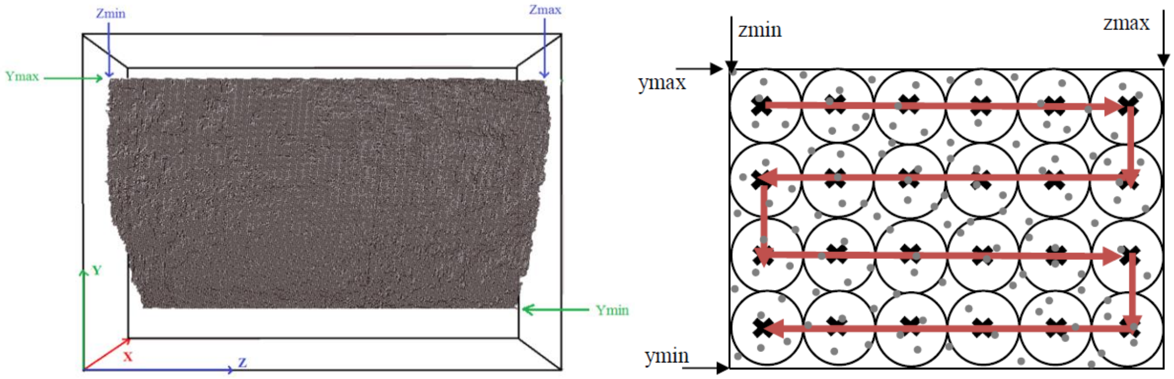
2) Trajectory planning

In order to follow the vehicle surface within a given region of interest, the trajectory of the end effector is planned using a raster scan motion according to the 3D model defined in the previous section. The global path planning strategy, introduced in [24], assumes that each zone of interest is bounded in a rectangular box delimited by a set of points at the edge of the surface (Y_{min}, Y_{max}, Z_{min}, Z_{max}), corresponding respectively to the minimum and maximum values of the Y and Z coordinates in the vertex list (Fig. 10a). To scan and explore the whole region, the end-effector center position at each step is defined by a set of points according to the end-effector coverage zone, which corresponds to the size of the tool attached on the robot to perform the screening operation (Fig. 10b). In this work the manipulator is considered to cover a circular area with radius of r = 50 mm which represents the diameter of the tool mounted on the robot to perform particles collection which are analyzed later on to detect the presence of any prohibited substances.

The planning strategy starts with the closest vertex to the upper-left corner of the rectangular bounding box. For this purpose the closest point to the corner point with Y_{max} and Z_{min} (Fig. 10b) coordinates is searched in the vertex list (Eq. 8).

$$\begin{gathered} \textit{for } (k=0{:}i_{num}) \\ \textit{Min(Dist)}= \\ \min\left(\sqrt{|\,VL[i].y - Ymax|^2 - \,|\,V[i].z - Zmin|^2}\right) \end{gathered} \tag{8}$$

where i_{num} is the total number of vertices in vertex list, and $VL[i]$ is an entry in the list of vertices. The vertex which generates the minimum distance is considered as start point. Then the robot moves horizontally towards the next point that is within a distance *2r*, which corresponds to twice the circular area of radius ***r*** covered by the end effector on the manipulator, from the previous point along the Z-axis until it reaches to the surface edge on the right side of the rectangular bounding box (Z_{max}). The surface edge is identified by checking the neighbour vertex at each step. Once the neighbour vertices are no longer within the rectangular area, the robot moves vertically down and the next position of the end-effector is the point which is within a distance *2r* from the previous location of the robot along the Y-axis. Then the robot again moves horizontally but in the opposite direction until it reaches the surface edge on the left side of the rectangular bounding area (Z_{min}). The process continues until the robot has scanned the entire area.



(a) (b)

Figure 10. a) Zone of interest bounded in a rectangular form, b) trajectory planning over the region of interest.

However, the desired region of interest can be of any shape and it is not efficient to scan a region only using a rectangular bounding approach. To further increase the flexibility and efficiency of the path planning such that it follows the actual edges of the region of interest, it is required to determine the object boundary points along the desired path. For this purpose, the surface is divided by *M* horizontal lines separated by a distance of *2r*, one from each other, along the Y axis (Fig. 11a).

$$M = (Y_{max} - Y_{min})) \ / \ 2r \qquad (9)$$

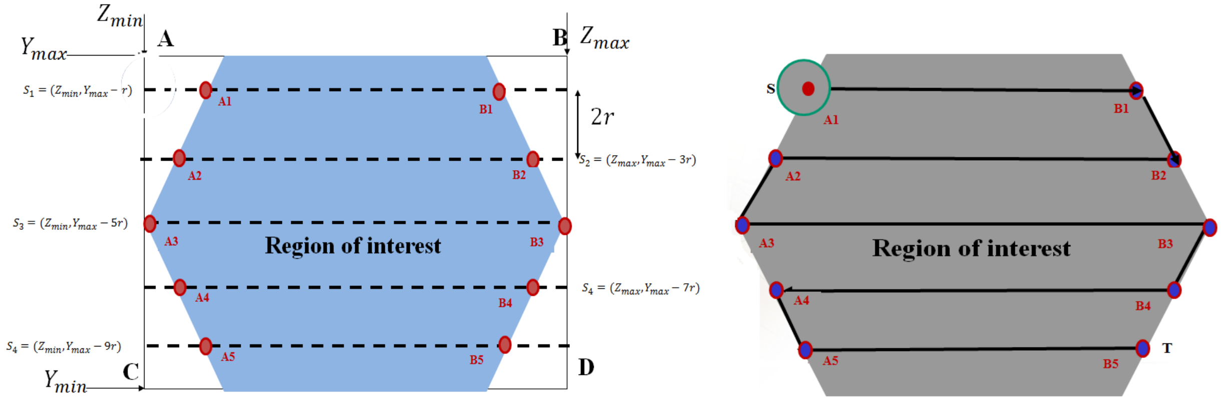
After splitting the surface, the distance between the points which lie on a same line and the line's start point is calculated. The points with the minimum (A_i) and maximum (B_i) distances are selected respectively as the border edges (Fig. 11b).

$$\begin{gathered} For(i = 0{:}\,M) \\ S_i = \ Y_{max} - (2i - 1)r \end{gathered} \qquad (10)$$

$$\begin{gathered} If \ |VL[i].y \ - S_i| < \ \alpha \\ Dist = \ |VL[i].z - Z_{min}| \end{gathered} \qquad (11)$$

$$\begin{gathered} \mathrm{A}_i \ = \ \min\,(Dist) \\ \mathrm{B}_i \ = \ \max\,(Dist) \end{gathered} \qquad (12)$$

where α is a small value to find the points that lies on the same horizontal line. These steps result in extracting all border edges with various depths. These edge points are used to control the robot position at the beginning and end of each horizontal line.


Z_{min}
Y_{max}
A
B
Z_{max}
$S_1 = (Z_{min}, Y_{max} - r)$
A1
B1
$2r$
$S_2 = (Z_{max}, Y_{max} - 3r)$
A2
B2
$S_3 = (Z_{min}, Y_{max} - 5r)$
A3
B3
Region of interest
$S_4 = (Z_{max}, Y_{max} - 7r)$
A4
B4
$S_4 = (Z_{min}, Y_{max} - 9r)$
A5
B5
C
D
Y_{min}
S
A1
B1
A2
B2
A3
Region of interest
B3
A4
B4
T
A5
B5

(a) (b)

Figure 11. a) Region of interest splitting, b) edge point's extraction.

3) Normal to surface calculation

The set of points forming a path for the robot defined in the previous step determine the positions for the end effector to reach in Cartesian coordinates. However, to ensure proper alignment of the end effector with the surface and close proximity to the vehicle bodywork, the local orientation of the end effector should match that of the normal to the surface of the vehicle at every point. This is especially important given the typical aesthetic curves and significant changes of orientation that exist over any type of vehicle. Therefore, it is required to calculate the normal to the surface and to estimate the local surface orientation in order to compute the proper end-effector orientation and ensure precise surface following. As defined earlier, using the vertex list and the face list, a triangle mesh of the surface is formed. The normal of each triangle is computed as the cross product between the vectors representing two edges of the triangle. The probability of having a normal in one direction is the same as having it in the opposite direction, which depends on the cross product order (Fig. 12a). To solve this issue, the direction of the normal is imposed to be in the same direction for all triangles, which corresponds to the orientation of the surface of the vehicle when viewed from the sensors side. The following equations define the normal vector, *N*, calculated from a set of three vertices, *E*, *F*, *G*, belonging to a given triangle:

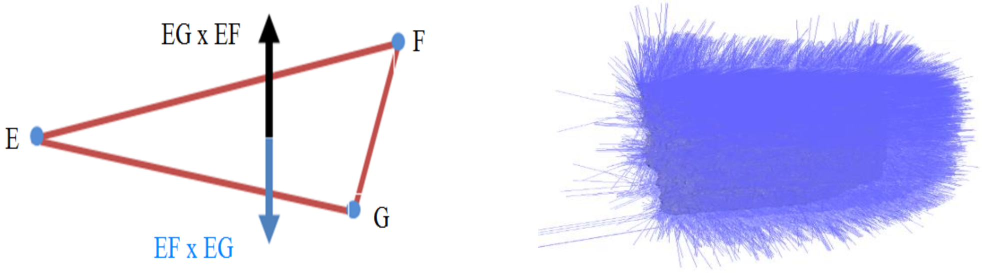
$$N_x = (E_y - G_y \cdot E_z - F_z) - (E_z - G_z \cdot E_y - F_y) \quad (13)$$

$$N_y = (E_z - G_z \cdot E_x - F_x) - (E_x - G_x \cdot E_z - F_z) \quad (14)$$

$$N_z = (E_x - G_x \cdot E_y - F_y) - (E_y - G_y \cdot E_x - F_x) \quad (15)$$

$$N = [N_x \quad N_y \quad N_z] \quad (16)$$

However, as shown in Fig. 12b, the data collected by Kinect sensors tend to be noisy and discretized. As a consequence, the normal calculation tends not to provide accurate enough information to properly determine the orientation in all locations.



(a) (b)

Figure 12. a) Triangle normal vector calculation, b) triangle mesh normals over a region of interest.

Therefore, the resulting normals are then normalized such that the length of the edges does not come into account. The normalized vector *N_{norm}* is computed as:

$$N_{norm} = \begin{bmatrix} \frac{N_x}{norm} & \frac{N_y}{norm} & \frac{N_z}{norm} \end{bmatrix} \tag{17}$$

where

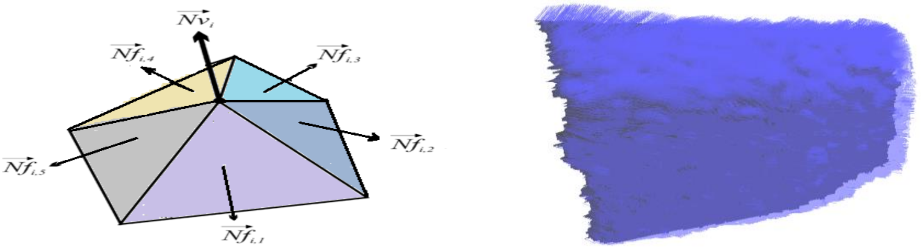
$$norm = \sqrt{N_x^2 + N_y^2 + N_z^2} \tag{18}$$

Since the planned path is a set of vertices from the vertex list, it is desired to calculate an orientation corresponding to each vertex. As shown in Fig. 13a, each vertex can be shared with several triangles. Therefore, to calculate the vertex normal in the triangle mesh, all the triangles that contain that vertex must be known and the normalized average of all these triangles' normal represents the vertex normal. If $\boldsymbol{f_{i,j}}$ are the *n* triangles that have the common vertex $\boldsymbol{V_i}$, the vertex normal $\overrightarrow{\boldsymbol{N_{v_i}}}$ is given as follows:

$$\overrightarrow{N_{v_i}} = \frac{\sum_{j=1}^{n} \overrightarrow{N_{f_{i,j}}}}{n} \tag{19}$$

In order to smooth out the normal estimate, the vertex's normal is calculated for all the local vertices which are covered by the end-effector, within a region of radius *r*, for every displacement step. The average of the vertices normals is then calculated and normalized to represent the overall direction of the surface in the local neighborhood of the corresponding vertex along the planned trajectory (Fig.13b).

The normal vector is then used to calculate a rotation matrix and compute the RPY angles that define the surface orientation with respect to the robot base frame. Since the vehicle panel surface and the robot end effector face directions that are opposite to each other, the orientation of the end effector for a given location along the trajectory is set as the opposite to the local surface orientation vector.

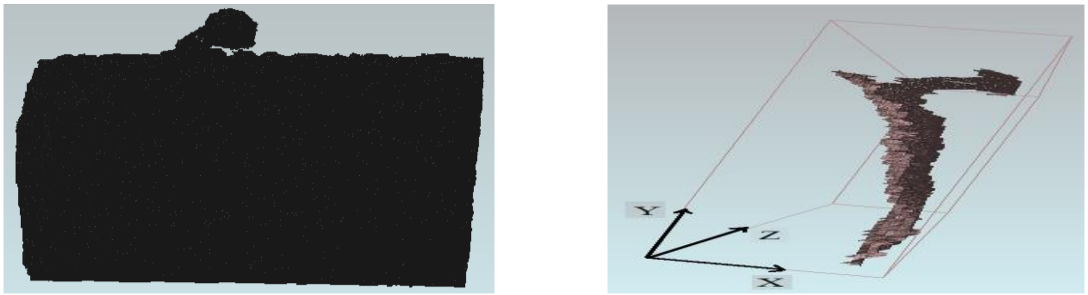

$\overrightarrow{Nv_i}$
$\overrightarrow{Nf_{i,4}}$
$\overrightarrow{Nf_{i,3}}$
$\overrightarrow{Nf_{i,2}}$
$\overrightarrow{Nf_{i,5}}$
$\overrightarrow{Nf_{i,1}}$

(a) (b)

Figure 13. a) Vertex normal calculation, b) averaged and normalized vertex normal.

4) Obstacle detection and avoidance

The desired path is safe as long as there is no obstacle (e.g. protruding components of the vehicle) along the planned path, as exemplified in Fig. 14a where the rear view mirror on the door panel is evidenced. In order to avoid an obstacle, it is required to detect the object using the 3D model computed in previous stages and define a new path for the robot in proximity of the obstacle. In this work, the obstacle detection is based on the detection of depth change in the local surface mesh acquired from the RGB-D sensors. A large and abrupt change in the depth value (x-coordinate) of a series of the vertices indicates the presence of an obstacle (Fig. 14b). For this purpose, the depth of the current position of the end effector is compared with the one in the next position. When the difference between the current and the next depth is greater than a threshold, here set to $\varepsilon_d = 30\,mm$, for at least 50 consecutive vertices, an obstacle is detected on the next position. In this case, the end effector moves backward from the predefined surface and temporarily follows the obstacle boundary until the difference between the depth of the current and the next position becomes lower than ε_d. It then moves forward to continue along the path originally planned unless another obstacle is detected.



(a) (b)

Figure 14. a) 3D model of a car door with a rear view mirror, b) abrupt depth change in presence of an obstacle.

5) Surface following execution

As a final stage, the computed Cartesian space trajectory which defines the position and orientation for the end-effector, as determined in the previous section, is transformed to the joint space of the robot via its inverse kinematics. Then, the joint space trajectory is sent to the controller to perform the surface following over the region of interest on the vehicle. The entire procedure is repeated for every region of interest extracted from the RGB-D model of the vehicle.

V. EXPERIMENTAL RESULTS

The algorithms used for this work were developed in C++ and run on a computer with an Intel core i7 CPU and Windows 7. For a 3D surface shape reconstruction of a regular vehicle, the average time needed is 4.0 seconds per Kinect sensor view. When using a network of 5 Kinect sensors covering an 180^o lateral view of the vehicle, the network of sensors collects the information in two time slots to avoid interference; the initialization of each device takes between 1 and 2 seconds. As a result, the scanning and 3D textured modeling processes for the entire side of a vehicle are completed within 30 seconds. The calibration process is performed off-line and has no impact on the inspection rate. To validate the feasibility of the proposed path planning method and to assess the accuracy of the end-effector motion in accordance with the surface of automotive panels, experiments are carried out with a 7-DOF CRS F3 manipulator which consists of a 6-DOF robot arm mounted on a linear track. The resulting redundancy provides an additional axis of motion and an expanded working area which is also useful to prevent some singularities.

a. Surface following over a mock-up car door with variable orientation

In the first experiment, a 3D model of a car door (Fig. 15a) is constructed using the color and depth information collected with the network of Kinect sensors. It is desired to closely scan the whole door with the robot except for the window area since glass is not imaged well with Kinect sensor technology. The considered region of interest (contained within the black rectangular box in Fig. 15a) over the door is stored as a 3D triangular mesh (Fig.15b).

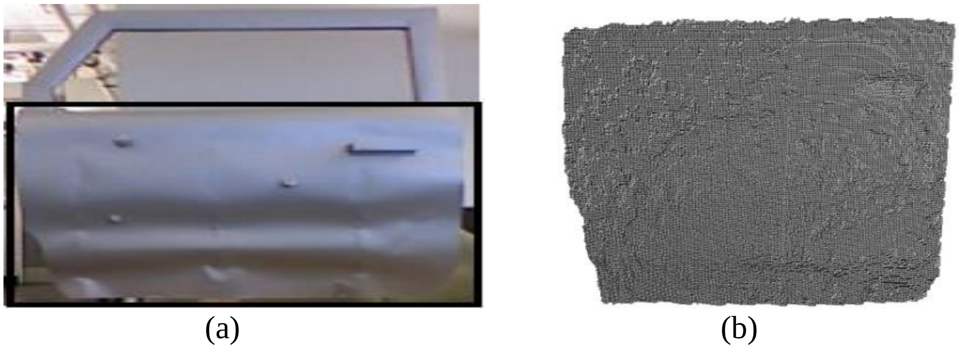
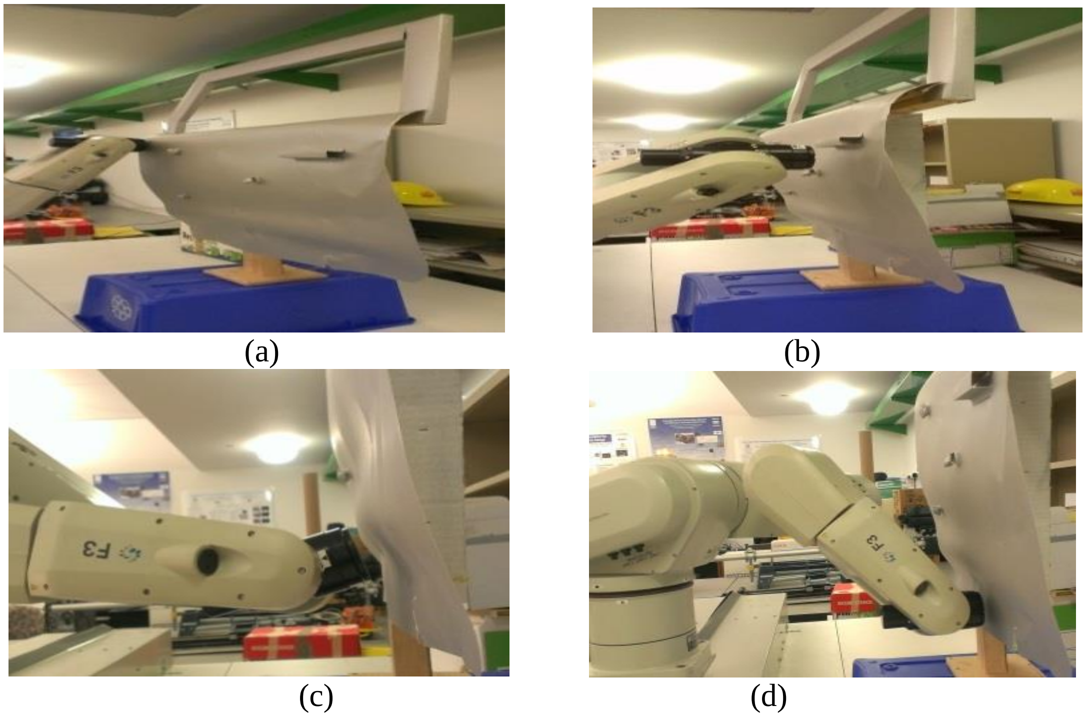

(a) (b)

Figure 15: a) Region of interest over a car door, b) corresponding 3D triangular mesh used to guide the manipulator.

The start point at the upper-left corner of the surface is detected (Fig. 16a). Then the robot follows the planned trajectory while remaining in close proximity to the surface (a safety distance of 50 mm is considered between the end-effector and the surface during this experiment). Over the entire trajectory, the end effector accurately follows the surface curves (Fig. 16b-d).

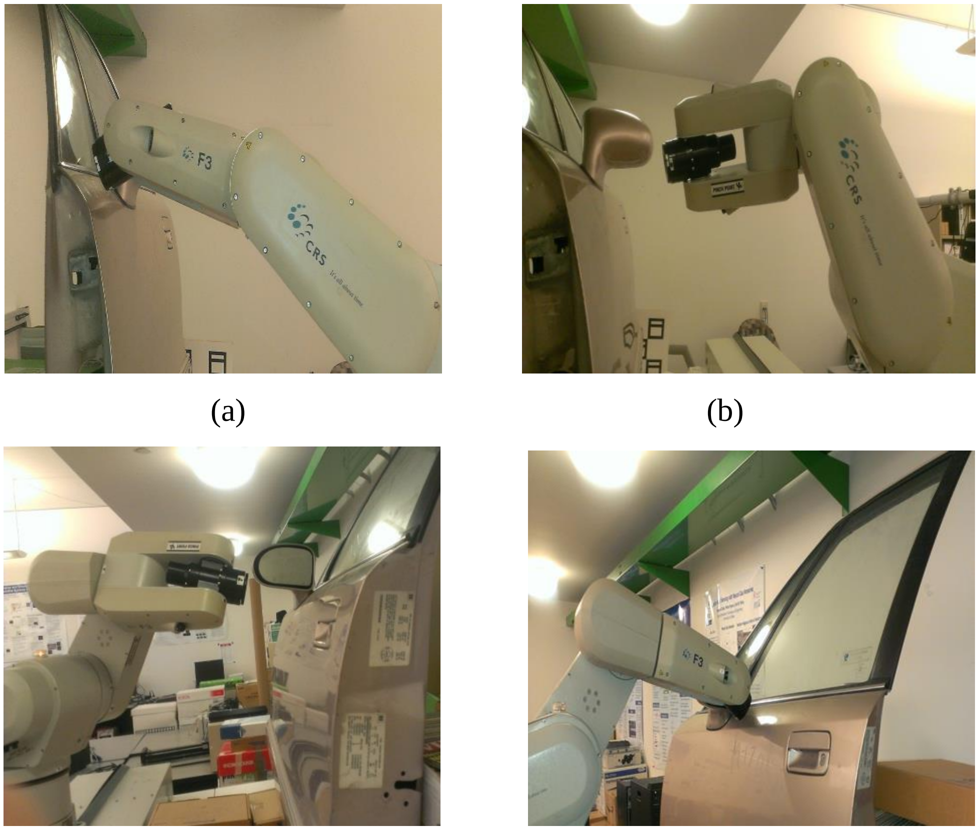

(a) (b)

(c) (d)

Figure 16: Robot performance at following the panel's curved surface and matching its position and orientation.

b. Surface following over a real car door with an obstacle

In the second experiment, a real automotive door panel with a rear view mirror is considered. As shown in Fig. 17a, a start point (upper left edge) of the object is automatically selected by the robot using the global information provided from the Kinect sensors. The obstacle (mirror) on the door is detectable in the reconstructed 3D model of the object (Fig. 14). The robot closely follows the object's surface until it reaches to the obstacle. Then the robot moves away from the panel, where the obstacle starts (Fig. 17b) and follows the obstacle boundary until the obstacle ends (Fig. 17c). Finally the robot moves back closer to the surface and continues closely following the surface on the other side of the obstacle (Fig. 17d).



(a) (b)

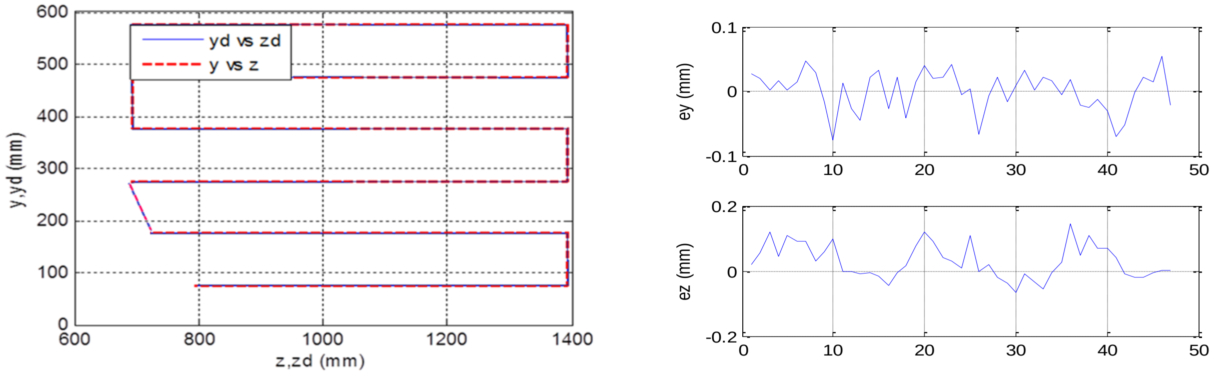
(c) (d)

Figure 17. Robot operation when dealing with an obstacle over the surface of a car door panel with a rear view mirror.

c. Surface following accuracy and limitations

Fig. 18a shows the robot trajectory in Cartesian space for the first experimental scenario above where the robot follows the surface of a mock-up car door (Fig. 16). In this case, the robot successfully tracks the desired path and scans the whole surface smoothly. The blue line shows the desired path (y_d, z_d) and the red dashed line shows the actual robot trajectory (y,z). It is observed that the robot path perfectly follows the desired trajectory along the surface of the panel. The desired path is planned based on the 3D model of the door collected by the vision sensors as shown in Fig. 15b, which ensures the proper alignment of the planned trajectory over the surface of the panels. Therefore, the error, (e_y, e_z), between the desired path and the robot trajectory (Fig. 18b) represents the accuracy of robot in following the Cartesian path, which remains under 0.15 mm of deviation. Fig. 19a shows the desired value,q_d, (in blue) of the joints calculated using inverse kinematics at each step versus the real joints value, q, (in red) after execution of each step. As shown in Fig. 19b, the error in joint space is also very small. Despite the low resolution on depth data collect by Kinect sensors over the region of interest of a car door, the surface is covered entirely and the experimental results show an adequate surface following performance in the joint and Cartesian spaces.

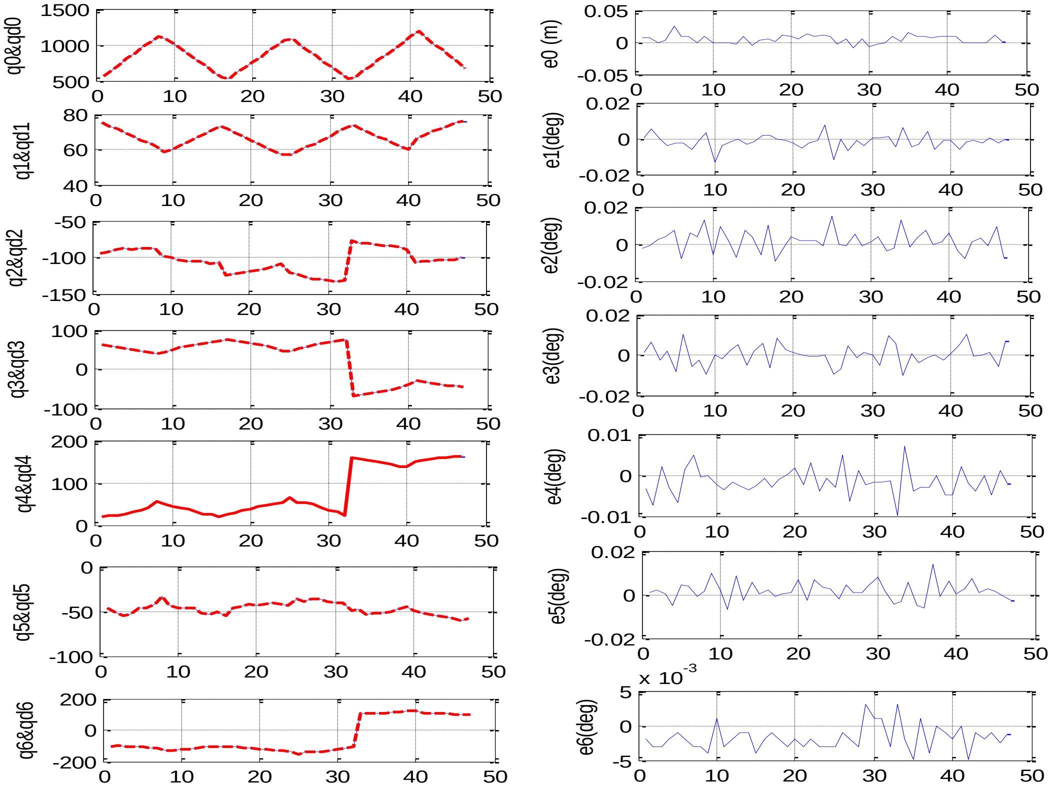
Beside the considerable advantages of using Kinect sensors, such as their acquisition speed and affordability, they also present some limitations, mainly related to their limited depth resolution. In order to further evaluate the potential error between the planned path based on the 3D model of an object and the actual position and shape of the panel in 2D (Y-Z plane), a refined contour following method was also evaluated [12] where an eye-in hand camera is mounted on the robot end-effector to provides supplementary local information with higher resolution about the actual contour location. A contour tracking operation is here considered as an alternative to surface following.



(a) (b)

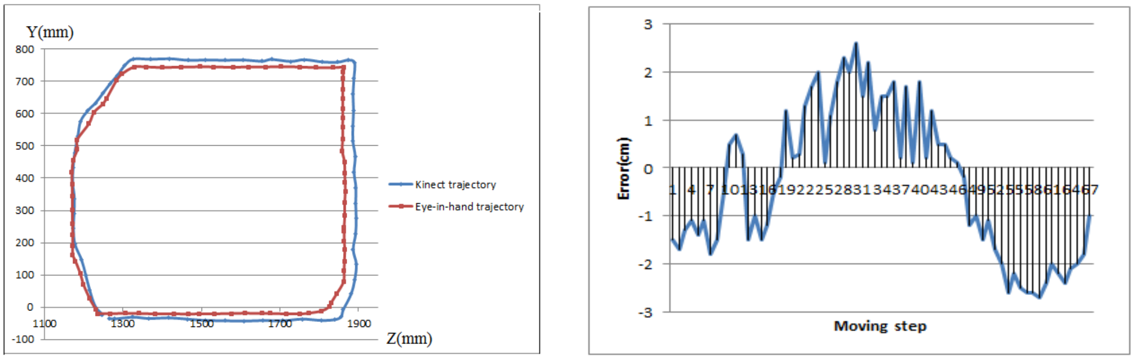
Figure 18: a) Cartesian space surface following trajectory, b) tracking error in the Cartesian space.

Fig. 20a shows the trajectory of the robot while following the contours of the automotive door shown in Fig. 15a, using Kinect information (blue edge map) vs the robot trajectory using visual servoing with aid of the eye-in-hand camera (red dots) to closely follow the contours. The gap in between the blue and the red contour points shows that the error between the estimated contour using Kinect sensors and the actual position of the object's contour is at most 30 mm. The error on the depth measurement provided by Kinect sensors during the first experiment is also shown in Fig. 20b. The Cartesian distance deviation that appears in between the end-effector location and the surface, was manually measured at each step over the surface following operation. The average residual error is about 20 mm in absolute value, and varies in sign depending on the section of the object and its distance from the Kinect. This level or error is expectable as it corresponds to the typical error characteristics of the Kinect sensors and of some residual error in the Kinect/robot calibration process discussed in section IV.b.



(a) (b)

Figure 19: a) Joint space surface following, and b) tracking error in the joint space.



(a) (b)

Figure 20: a) Car door contour following using an assistive eye-in-hand camera and deviation with respect to Kinect only guided navigation, b) depth error estimation for surface following in the first experiment.

VI. CONCLUSION AND FUTURE WORK

This work presents the design and implementation of an RGB-D vision-guided surface following method for automated and rapid vehicle inspection using the 3D modeling of regions of interest over an automotive vehicle in the context of automated security screening for restricted areas access. A layout for the vehicle screening station is proposed and the development of the main stages of the inspection system is detailed as well as their integration into a coordinated and efficient system solution able to meet tight time requirements. An automated classification technique is adapted for recognizing different vehicle categories from piecewise views collected by a calibrated network of Kinect sensors distributed around the vehicle. The sensors collect color and depth information over the bodywork of the vehicle to produce a triangular mesh in an accelerated manner. The resulting 3D data provides sufficiently accurate spatial information about the bodywork of the vehicle for planning the path of a robot to closely follow the surface of an automotive vehicle surfaces as required to collect samples of particles that support the detection of prohibited substances that may be hiding on board the vehicle. The proposed method is implemented and experimentally validated at full scale with a network of five RGB-D sensors and a 7-DOF robotic manipulator. The experimental results demonstrate that sufficient accuracy is achieved for the robot to successfully follow, both in position and orientation, the surface and the contours of whole regions of interest, while also avoiding obstacles created by protruding parts over the vehicle body.

Given the noisy and discretized nature of information provided by Kinect sensors, especially on depth measurements, ensuring and maintaining contact with the vehicle during the entire screening process remains a challenge. To address this issue an adaptive path planning method is currently being developed to further refine the planned trajectory based on extra information about the local shape and orientation of the surface. The required information will be collected using live proximity and contact sensors mounted on the end-effector to dynamically refine the planned trajectory. Any error due to the relatively low accuracy of the 3D model of the vehicle achieved with the RGB-D sensors will be compensated by these embedded sensors in order to ensure more precise interaction between the robot and the vehicle.

ACKNOWLEDGEMENTS

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada via the Strategic Project Grant no. STPGP 381229-09, and performed in collaboration with Scintrex Trace Corp.

REFERENCES

[1] Y.H. Anis, M.R. Holl and D.R. Meldrum, “Automated Selection and Placement of Single Cells Using Vision-Based Feedback Control”, IEEE Transactions on Automation Science and Engineering, vol. 7, no. 3, July 2010, pp. 598-606.

[2] R. Araujo, U. Nunes and A. T. de Almeida, "Robot 3D Force-Based Surface-Tracking," 20th International Conference on Industrial Electronics, Control and Instrumentation, vol. 2, 1994, Bologna, pp. 788-793.

[3] Y. Yin, H. Hu and Y. Xia, “Active Tracking of Unknown Surface Using Force Sensing and Control Technique for Robot”, Sensors and Actuators, vol.112, 2004, pp. 313–319.

[4] X. Papageorgiou, H. G. Tanner and K. J. Kyriakopoulos, "Motion Tasks for Robot Manipulators on Embedded 2-D Manifolds Under Input Constraints," Control Conference (ECC), 2007 European, Kos, 2007, pp. 3783-3789.

[5] X. Papageorgiou, S. G. Loizou and K. J. Kyriakopoulos, "Motion Tasks and Force Control for Robot Manipulators on Embedded 2-D Manifolds," IEEE International Conference on Robotics and Automation, Roma, 2007, pp. 4202-4207.

[6] R.S. Dahiya, G. Metta, G. Cannata and M. Valle, “Guest Editorial: Special Issue on Robotic Sense of Touch,” IEEE Transactions on Robotics, vol. 27, no. 3, 2011, pp. 385–388.

[7] R. Ibrayev and Y.B. Jia, “Recognition of Curved Surfaces from “One-dimensional” Tactile Data,” IEEE Transactions on Automation Science and Engineering, vol. 9, no. 3, 2012, pp. 613-621.

[8] J.L. Pedreño-Molina, A. Guerrero-González, J. Calabozo-Moran, J. López-Coronado and P. Gorce, “A Neural Tactile Architecture Applied to Real-Time Stiffness Estimation for a Large Scale of Robotic Grasping Systems,” Journal of Intelligent and Robotic Systems, vol. 49, 2007, pp. 311-323.

[9] Y. Ito, Y. Kim, C. Nagai and G. Obinata,"Vision-Based Tactile Sensing and Shape Estimation Using a Fluid-Type Touchpad," IEEE Transactions on Automation Science and Engineering, vol. 9, no. 4, Oct. 2012, pp. 734-744.

[10] J. Wang and Y. Li, “Surface-Tracking of a 5-DOF Manipulator Equipped with Tactile Sensors,” 11th International Conference on Control Automation Robotics & Vision, 2010, Singapore, pp. 2448- 2453.

[11] J. Wang and Y. Li, “Tracking Control of a Redundant Manipulator with the Assistance of Tactile Sensing,” Intelligent Automation and Soft Computing, vol. 17, no. 7, 2011, pp. 833-845.

[12] D. Nakhaeinia, P. Payeur and R. Laganière, "Adaptive Robotic Contour Following from Low Accuracy RGB-D Surface Profiling and Visual Servoing," Conference on Computer and Robot Vision (CRV), Montreal, 2014, pp. 48-55.

[13] A. Pichler and M. Jagersand, "Uncalibrated Hybrid Force-Vision Manipulation," IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS 2000), vol. 3, Takamatsu, 2000, pp. 1866-1871.

[14] J. Baeten, H. Bruyninckx and J. D. Schutter, “Integrated Vision/Force Robotic Servoing in the Task Frame Formalism,” The International Journal of Robotics Research, vol. 22, 2003, pp. 941–954.

[15] E. Li, Z. Li and J. He, “Robotic Adaptive Impedance Control Based on Visual Guidance,” Int. Journal Smart Sensing and Intelligent Systems, vol. 8, no 4, December 2015, pp. 2159-2174.

[16] C.C. Cheah, S.P. Hou, Y. Zhao and J.J.E. Slotine, “Adaptive Vision and Force Tracking Control for Robots with Constraint Uncertainty”. IEEE/ASME Transactions on Mechatronics, vol. 15, no. 3, 2010, pp. 389–399.

[17] D. Nakhaeinia, P. Laferrière, P. Payeur, R. Laganière, “Safe Close-Proximity and Physical Human-Robot Interaction Using Industrial Robots,” Conference on Computer and Robot Vision (CRV), Halifax, 2015, pp. 237-244.

[18] R. Fareh, P. Payeur, D. Nakhaeinia, R. Macknojia, A. Chávez-Aragón, A.M. Cretu, P. Laferrière, R. Laganière and R. Toledo, “An Integrated Vision-Guided Robotic System for Rapid Vehicle Inspection,” IEEE International Systems Conference, Ottawa, 2014, pp. 446 - 451.

[19] C. Pillajo and J.E. Sierra, “Human Machine Interface HMI Using Kinect Sensor toC a SCARA Robot,” IEEE Colombian Conference on Communications and Computing, Medellin, 2013, pp. 1-5.

[20] A. Teichman, J.T. Lussier and S. Thrun, “Learning to Segment and Track in RGBD,” IEEE Transactions on Automation Science and Engineering, vol. 10, no. 4, 2013, pp. 841-852.

[21] F. Nagata, Y. Kusumoto, K. Watanabe, K. Kiguchi, K. Tsuda, K. Yasuda, K. Yokoyama, M. Umetsu, N. Mori and M. Omoto, "High Precision Polishing Robot Using a Learning-Based Surface Following Controller," IEEE International Symposium on Computational Intelligence in Robotics and Automation, vol.1, 2003, pp. 91-96.

[22] J. J. Park, J.H. Kim and J.B. Song, “Path Planning for a Robot Manipulator Based on Probabilistic Roadmap and Reinforcement Learning,” International Journal of Control, Automation, and Systems, vol. 5, no. 6, 2007, pp. 674-680.

[23] B. Hamner, S. Singh, S. Scherer, “Learning Obstacle Avoidance Parameters from Operator Behavior,” Journal of Field Robotics, Special Issue on Machine Learning Based Robotics in Unstructured Environments, vol. 23, no. 11-12, 2006, pp. 1037–1058.

[24] D. Nakhaeinia, R. Fareh, P. Payeur and R. Laganière, “Trajectory Planning for Surface Following with a Manipulator Under RGB-D Visual Guidance. IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), Linkoping, Sweden, 2013, pp. 1-6.

[25] R. Macknojia, A. Chávez-Aragón, P. Payeur and R. Laganière, “Calibration of a Network of Kinect Sensors for Robotic Inspection Over a Large Workspace,” IEEE Workshop on Robot Vision, Clearwater, FL, 2013, pp. 184-190.

[26] A.-M. Cretu, and P. Payeur, "Biologically-Inspired Visual Attention Features for a Vehicle Classification Task", Int. Journal Smart Sensing and Intelligent Systems, vol. 4, no. 3, Sep. 2011, pp. 402-423.

[27] L. Shui, “Adaptive Image Segmentation Based on Saliency Detection,” Int. Journal Smart Sensing and Intelligent Systems, vol. 8, no. 1, March 2015, pp. 408-428.

[28] L. Itti, C. Koch and E. Niebur, “A Model of Saliency-Based Visual Attention for Rapid Scene Analysis,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.20, no.11, 1998, pp. 1254–1259.

[29] A. Chávez-Aragón, R. Macknojia, P. Payeur and R. Laganière, “Rapid 3D Modeling and Parts Recognition on Automotive Vehicles Using a Network of RGB-D Sensors for Robot Guidance” Journal of Sensors, Hindawi (ed), 2013, 16 pages.

[30] P. Lindstrom, “Out-of-Core Simplification of Large Polygonal Models,” 27th Annual Conference on Computer Graphics and Interactive Techniques, New York, 2000, pp. 259-262.