Hindawi Publishing Corporation
Journal of Sensors
Volume 2013, Article ID 832963, 16 pages
http://dx.doi.org/10.1155/2013/832963

Research Article

Rapid 3D Modeling and Parts Recognition on Automotive Vehicles Using a Network of RGB-D Sensors for Robot Guidance

Alberto Chávez-Aragón, Rizwan Macknojia, Pierre Payeur, and Robert Laganière

Faculty of Engineering, School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, Canada K1N 6N5

Correspondence should be addressed to Alberto Chávez-Aragón; achavez@uottawa.ca

Received 15 March 2013; Revised 9 July 2013; Accepted 12 July 2013

Academic Editor: Lei Wang

This paper presents an approach for the automatic detection and fast 3D profiling of lateral body panels of vehicles. The work introduces a method to integrate raw streams from depth sensors in the task of 3D profiling and reconstruction and a methodology for the extrinsic calibration of a network of Kinect sensors. This sensing framework is intended for rapidly providing a robot with enough spatial information to interact with automobile panels using various tools. When a vehicle is positioned inside the defined scanning area, a collection of reference parts on the bodywork are automatically recognized from a mosaic of color images collected by a network of Kinect sensors distributed around the vehicle and a global frame of reference is set up. Sections of the depth information on one side of the vehicle are then collected, aligned, and merged into a global RGB-D model. Finally, a 3D triangular mesh modelling the body panels of the vehicle is automatically built. The approach has applications in the intelligent transportation industry, automated vehicle inspection, quality control, automatic car wash systems, automotive production lines, and scan alignment and interpretation.

1. Introduction

Robot manipulation and navigation require efficient methods for representing and interpreting the surrounding environment. Industrial robots, which work in controlled environments, are typically designed to perform only repetitive and preprogrammed tasks. However, robots working in dynamic environments demand reliable methods to interpret their surroundings and are submitted to severe time constraints. Most existing solutions for robotic environment representation and interpretation make use of high-cost 3D profiling cameras, scanners, sonars, or combinations of them, which often result in lengthy acquisition and slow processing of massive amounts of information. The extreme acquisition speed of the Kinect's technology meets requirements for rapidly acquiring models over large volumes, such as that of automotive vehicles. The performance, affordability, and the growing adoption of the Kinect for robotic applications supported the selection of the sensor to develop the robotic inspection station operating under multisensory visual guidance. The method presented in this work uses a set of Kinect depth sensors properly calibrated to collect visual information as well as 3D points from different regions over vehicle bodyworks. A dedicated calibration methodology is presented to achieve accurate alignment between the respective point clouds and textured images acquired by Kinect sensors distributed in a collaborative network of imagers to provide coverage over large surfaces. First, the sensing system uses computer vision and machine learning techniques for determining the location and category of a vehicle and some areas of interest over the bodywork. Then, the 3D readings are aligned using the extrinsic parameters between the Kinect units. Finally, a 3D triangle mesh, modeling the lateral panels of the vehicle, is built and serves as input to guide a manipulator robot that will interact with the surface. The experiments reported in this work are the result of processing images and point clouds of side panels of automobiles. Nevertheless, the method can be adapted easily to recognize other types of objects.

This work contributes to the robotic vision field by proposing a simple and efficient methodology for automatic 3D surface modeling of large vehicle parts via the coordinated and integrated operation of several RGB-D sensor heads; a dedicated methodology for extrinsic calibration of Kinect sensors, as well as a rapid algorithm for triangle meshing

which takes advantage of the structure of the point clouds provided by the Kinect sensors.

2. Related Work

In the recent years the Kinect device has been widely adopted as an indoor sensor for robotics and human-computer interaction applications. The sensor is a multiview structured lighting system, containing an RGB camera, an infrared (IR) camera, and an infrared laser projector equipped with a microgrid that artificially creates a predefined IR pattern over the imaged surface. The sensor is capable of collecting depth information for each pixel in a color image, which opens the door to a great variety of applications. Lately, two dominant streams of research have been pursued with Kinect technology: (1) the investigation of the technology behind the device, analysis of its properties, performance, and comparison with other depth sensors; (2) the development of applications of the Kinect technology in fields such as robotics, user interfaces, and medicine among others. The present work addresses both categories.

Among the numerous examples of applications for the Kinect technology that rapidly appeared in the literature, Zhou et al. [1] proposed a system capable of scanning human bodies using multiple Kinect sensors arranged in a circular ring. Maimone and Fuchs [2] presented a real-time telepresence system with head tracking capabilities based on a set of Kinect units. They also contributed an algorithm for merging data and automatic color adjustment between multiple depth data sources. An application of Kinect in the medical field for position tracking in CT scans was proposed by Noonan et al. [3]. They tracked the head of a phantom by registering Kinect depth data to high resolution CT template of a head phantom. Rakprayoon et al. [4] used a Kinect sensor for obstacle detection of a robotic manipulator. In [5], Berger et al. originally used multiple Kinect sensors for aerodynamic studies of 3D objects. They captured and visualized gas flow around objects with different properties. Smisek et al. [6] and Park et al. [7] conducted analyses regarding Kinect's depth resolution, accuracy with stereo resolution reconstruction, and camera calibration as well as a comparison with a laser scanner. For simultaneous calibration of the Kinect sensor, different approaches have been proposed. Burrus [8] proposed to use traditional techniques for calibrating the Kinect color camera and manual selection of the corners of a checkerboard for calibrating the depth sensor. Gaffney [9] described a technique to calibrate the depth sensor by using 3D printouts of cuboids to generate different levels in depth images. The latter, however, requires an elaborate process to construct the target. Berger et al. [10] used a checkerboard where black boxes were replaced with mirroring aluminium foil therefore avoiding the necessity of blocking the projector when calibrating the depth camera.

With regard to the depth data of the Kinect sensor, it is known that it suffers from quantization noise [6, 11] that increases as the distance to the object increases. The resolution also decreases with the distance [11]. The depth map may also contain occluded and missing depth areas mainly due to the physical separation between the IR projector and the IR camera and to the inability to collect sufficient IR signal reflection over some types of surfaces. These missing values can however be approximated by filtering or interpolation [2, 12].

Concerning the automated detection of vehicle parts, a variety of computer vision systems have been developed in the past that aimed at detecting regions of interest in images of vehicles. Among popular applications in this field, the inspection of products on assembly lines stands out. Some of these systems used methods to simultaneously locate many reference points or many regions of interest [13, 14]. To manage the semantic information in the problem domain, Kiryakov et al. [15] used templates and similarity measures to evaluate the correct position of a template over an image. For the visual detection of features of interest in images some authors have reported the successful use of a technique proposed by Viola and Jones called cascade of boosted classifiers (CBC) [16]. This technique has proven to be useful in detecting faces, wheels, back views of cars, and license plates among others [17, 18]. While applications of previous research works are mainly in the area of intelligent transportation systems (ITS), these concepts can advantageously be transposed for applications in robotic guidance.

3. Proposed RGB-D Acquisition Framework

The work presented here aims at the integration of information from multiple RGB-D sensors to achieve fully automated and rapid 3D profiling of bodywork regions over automotive vehicles. The approach estimates the shape over selected regions to be reconstructed based on the detection of features of interest on vehicle body panels. Once the location of the regions of interest is known, the approach reconstructs the panels' shape using information provided by a set of Kinect sensors placed conveniently which collect visual and 3D information from the vehicle.

The final goal of the system being developed is to support the real-time navigation of a robotic arm in proximity of the vehicle in order to perform a series of tasks (e.g., cleaning, maintenance, inspection) while it is interacting with the vehicle surface. The work reported in this paper focuses mainly on the robotic vision stage.

Figure 1 shows various environments used to develop and test the proposed system. In Figure 1(a), an indoor laboratory environment is depicted where an actual piece of automotive bodywork was used for early development along with a mockup car door model. In Figure 1(b) a multilevel semioutdoor parking garage is shown where real full-size vehicles were imaged to validate the accuracy of calibration over the network of RGB-D sensors. The parking garage infrastructure prevented the direct sunlight from causing interference with the IR image components of the Kinect units. Natural light coming from windows and open walls as well as electric lamps lighted up the scene. The experiments demonstrated that these sources of light did not trouble the sensors' technology.

The layout of a vehicle scanning station is shown in Figure 2. Yellow lines delimit the area where the vehicle stops,

(a) (b)

Figure 1: Indoor and semioutdoor environments used to develop and test the proposed acquisition stage.

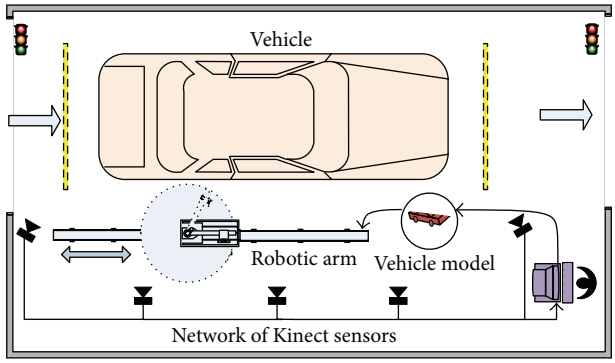


Figure 2: System layout of the proposed scanning system: yellow lines delimit the area where the vehicle stops, while depth and color information is collected.

Table 1: Parameters of the proposed scanning system.

	IR camera	RGB camera
Horizontal field of view	57°	63°
Vertical field of view	45°	50°
Distance between cameras	1.3 m	1.3 m
Height of the sensors above the ground	1 m	1 m
Distance between baseline cameras and vehicle	2 m	2 m
Horizontal overlapping area between two cameras	0.85 m	1.15 m
Coverage area for each sensor	4.7 m × 1.65 m	5 m × 1.85 m
Total coverage area for the baseline depth sensors	6 m × 1.65 m	6.3 m × 1.85 m

while depth and color information is collected. At the beginning, the robotic arm which can be moved on rails is positioned at the end of the track within the blind spots of the sensors. Then, the vehicle enters the scanning area and stops in the designated space. The Kinect sensors collect color and depth information over the entire length of the vehicle within 10 seconds. The information is then processed in order to construct a 3D model of the bodywork panels on the vehicle. The whole scanning and modeling process is meant to be fast, in order to support high inspection cadence. The latter criterion was the main factor to support adoption of the Kinect technology in this application in spite of its limited depth resolution and sensitivity to ambient lighting conditions.

Overall the scanning and 3D textured modeling processes are completed in 30 seconds, which rapidly makes the information available for robot guidance. The data processing pipeline of the sensory information provided by the Kinect devices is shown in Figure 3. Rounded rectangles represent the different modules that are part of the process. Those modules within the dotted rectangle operate on-line, while the calibration procedure out of the dotted rectangle is executed off-line and only once prior to inspection operation. Rectangles represent the inputs and output of the system.

3.1. System Configuration. The scanning system for collecting color and depth information over the whole lateral view of a vehicle bodywork consists in three Kinects positioned in a single row (sensor's baseline) with their viewing axes perpendicular to the bodywork and two extra sensors, which cover partially the front and back areas, rotated toward the vehicle in such a way that their viewing axes form a 65 degree angle with respect to the sensor's baseline.

This configuration can be replicated on the other side of the vehicle for a complete 360 degree view. As detailed in Table 1, the sensors are positioned at 1 m above the ground and parallel to the floor. The cameras were set up about 2 m away from the area where the vehicle is positioned. This configuration permits to meet the following requirements to cover the entire side of a car: (1) a minimum coverage area of 4.15 × 1.5 m, which is the typical size of a sedan vehicle; (2) collection of depth readings in the range of 0.8 to 3 m, which is the range where Kinect performs well; (3) an overlapping area in the range of 0.5 m to 1 m, between contiguous sensors to ensure accurate external calibration process and point cloud alignment. Figure 4 depicts the acquisition system. It is worth mentioning that the proposed acquisition system can be easily adapted for larger vehicles by including extra sensors in the sensor's baseline.

3.2. Interference Issues. Within the overlapping regions between two contiguous Kinect sensors, interference might happen between the sensors since all Kinect devices project a pattern of infrared points at the same wavelength to create their respective depth map. This produces small holes on the depth maps of overlapping sensors. To prevent this problem, the data is collected sequentially over different time slots. In the first time slot, sensors K_0 and K_2 simultaneously collect

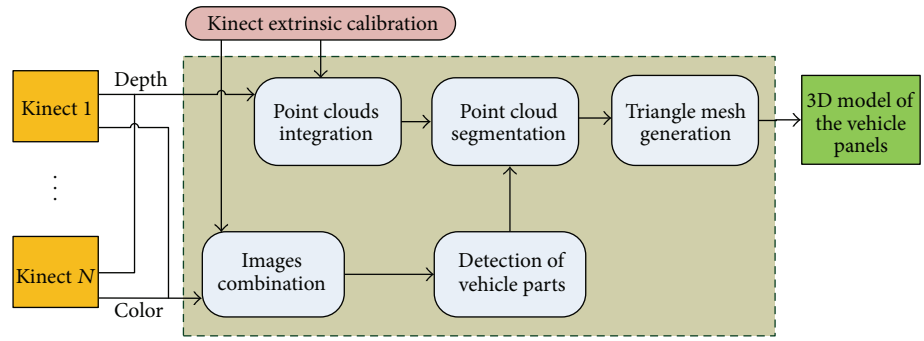


Figure 3: Main components of the RGB-D scanning and modeling approach.

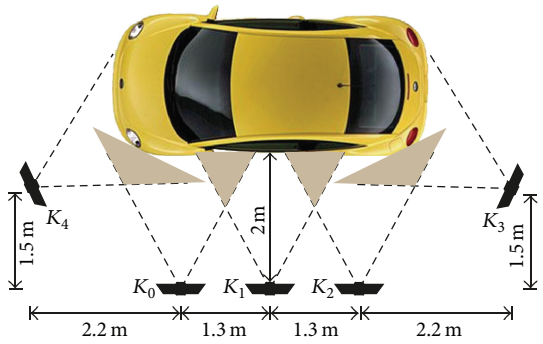


Figure 4: Configuration of the acquisition system. Five Kinects sensors are used to collect color and depth information over the entire set of lateral panels of a vehicle.

their respective information (see Figure 4). Then, sensors K_1, K_3, and K_4 scan the central, back, and front regions of the vehicle. The delay between the shots is the time needed to shut down the first group of devices and initialize the next one. This process is performed by the Kinect drivers and takes few seconds. The OpenNI framework was used to control the Kinect for Xbox 360 version of the sensors; this framework does not provide means to shut down only the infrared projector.

4. Calibration of a Network of Kinect Sensors

Kinect technology consists of a multiview system that provides three outputs: an RGB image, an infrared image, and a depth image for each sensor. Arranging a group of sensors as a collaborative network of imagers permits to enlarge the overall field of view and to model large objects, such as automotive vehicles. For an adequate operation of the network a precise mapping between color and infrared must be achieved. For this purpose an internal calibration procedure that estimates the intrinsic parameters of each camera within every device as well as the extrinsic parameters between the RGB and the IR camera inside a given Kinect is required, along with an external calibration process that provides accurate estimates of the extrinsic parameters in between the respective pairs of Kinect devices. A procedure dedicated to Kinect sensors is proposed and detailed later.

4.1. Internal Calibration

4.1.1. Intrinsic Parameters Estimation for Built-In Kinect Cameras. The internal calibration procedure includes the estimation of the respective intrinsic parameters for the color and the IR sensors, which are the focal length (f_x, f_y), the principal point (O_x, O_y), and the lens distortion coefficients $(k_1, k_2, p_1, p_2, k_3)$ [19]. Because the RGB and IR cameras exhibit different color responses, the proposed calibration technique uses a regular chessboard target of size 9×7 that is visible in both sensors' spectra. During internal calibration the Kinect's IR projector is blocked by overlapping a mask on the projector window. The IR projector otherwise introduces noise over the IR image as shown in Figure 5(a), and without projection, the image is too dark as shown in Figure 5(b). Therefore standard external incandescent lamps are added to illuminate the checkerboard target, Figure 5(c). The color image is not affected by the IR projection and creates a clear pattern, Figure 5(d).

The checkerboard was printed on a regular A3 size paper, which does not reflect back the bright blobs due to the external incandescent lamps in the IR image plane. To ensure the best calibration results, 100 images were collected from both the color and the IR cameras. Both images were synchronized in each frame, so that they could be used for extrinsic calibration between the cameras. To estimate the intrinsic parameters, each Kinect is calibrated individually using Zhang's camera calibration method [19]. The method is applied 10 times on 30 images randomly selected among the 100 captured images. The reprojection error is also calculated for each iteration, which is a measure of the deviation of the camera response to the ideal pinhole camera model. The reprojection error is calculated as the RMS error of all the target calibration points.

After calibration, both the RGB and IR cameras achieve reprojection error between 0.12 and 0.16 pixels, which is better than the original performance given by the default manufacturer calibration of the Kinect sensor. The reprojection error without calibration of the IR camera is greater than 0.3 pixel and that of the color camera is greater than 0.5 pixel. The focal length of the IR camera is larger than that of the color camera, that is, the color camera has a larger field of view. It is also apparent that every Kinect sensor has slightly different intrinsic parameters. This confirms the need for a formal

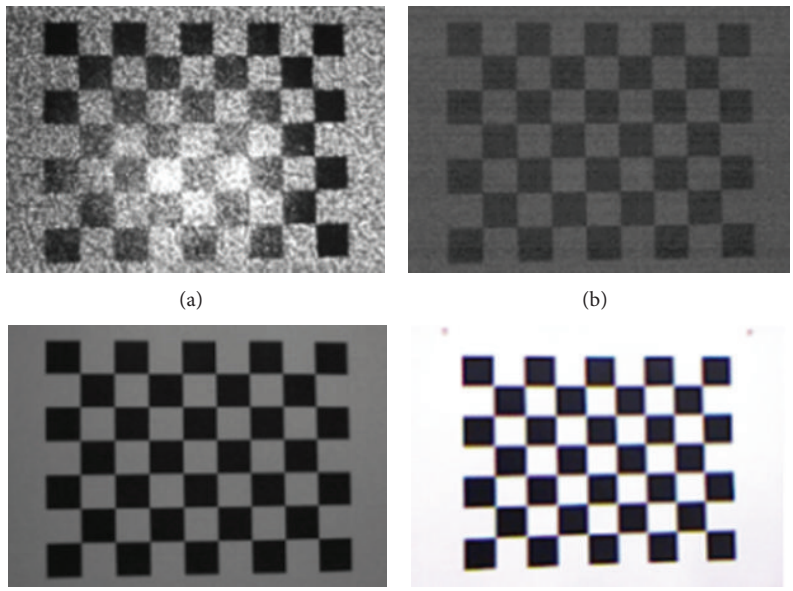


Figure 5: Views of the checkerboard in different configurations: (a) IR image with IR projector, (b) IR image without IR projector, (c) IR image with incandescent lighting and without projector, and (d) color image.

intrinsic calibration to be performed on each device to support accurate data registration.

4.1.2. Extrinsic Parameters Estimation for Built-In Kinect Cameras. The respective location of the color and IR cameras within each Kinect unit is determined by stereo calibration. The camera calibration method proposed by Zhang [19] also provides the location of the checkerboard target with respect to the camera coordinate system. If the target remains fixed for both cameras, then the position between both cameras is defined by (1)

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

where H is the homogenous transformation matrix (consists of 3×3 rotation matrix R and 3×1 translation vector T) from the RGB camera to the IR camera, H_{IR} is the homogenous transformation matrix from the IR camera to the checkerboard target, and H_{RGB} is the homogenous transformation from the RGB camera to the checkerboard target. The translation and rotation parameters between the RGB and IR sensors are shown in Table 2 for five Kinect sensors. The internal extrinsic calibration parameters allow to accurately relate the color to depth data collected by a given Kinect device.

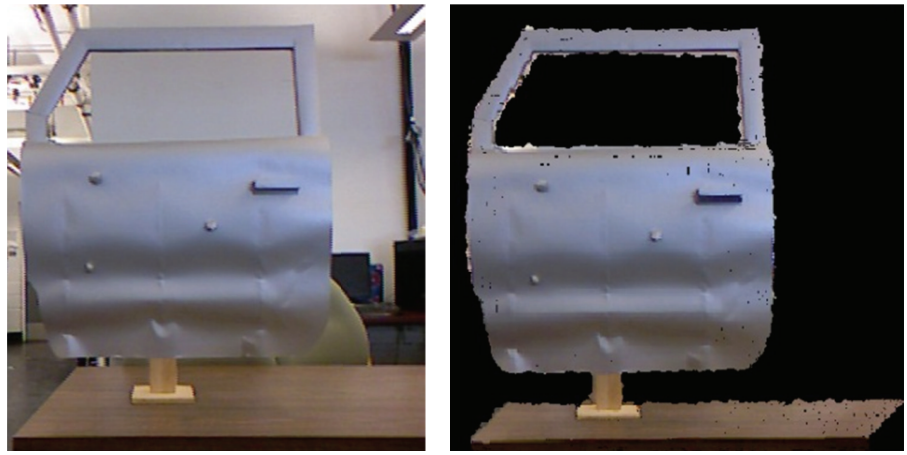
Table 2: Internal extrinsic calibration of embedded sensors.

Translation (cm) and rotation (degree) between RGB and IR						
Sensor	T_x	T_y	T_z	R_x	R_y	R_z
K_0	2.50	0.0231	0.3423	0.0017	0.0018	−0.0082
K_1	2.46	−0.0168	−0.1426	0.0049	0.0032	0.0112
K_2	2.41	−0.0426	−0.3729	0.0027	0.0065	−0.0075
K_3	2.49	0.0153	0.2572	−0.0046	0.0074	0.0035
K_4	2.47	0.0374	0.3120	0.0052	0.0035	0.0045

4.1.3. Registration of Color and Depth Information within a Given Kinect Device. The Kinect sensor does not provide the registered color and depth images. Once the internal intrinsic and extrinsic parameters are determined for a given Kinect device, the procedure to merge the color and depth based on the estimated registration parameters is performed as follows. The first step is to properly relate the IR image and the depth image. The depth image is generated from the IR image but there is a small offset between the two, which is introduced as a result of the correlation performed internally during depth calculation. The offset is 5 pixels in the horizontal direction and 4 pixels in the vertical direction [6]. After removing this offset using (2), each pixel of the depth image exactly maps the depth of the corresponding pixel in the IR image. Therefore, the calibration parameters of the IR camera can be applied on the depth image considered

$$dp(x, y) = ds(x - 5, y - 4), \tag{2}$$

where x and y are the pixel location, $ds(x, y)$ is the offsetted depth value affecting the Kinect depth sensor, and $dp(x, y)$ is the corrected depth value. The second step consists in transforming both the color and the depth images to compensate for radial and tangential lens distortion using OpenCV undistort function [20]. This function estimates the geometric transformation on the images using the distortion parameters and provides the undistorted color image and depth image ($du(x, y)$). The next step is to determine the 3D coordinates



(a) (b)

Figure 6: Accurate integration of color and depth images.

corresponding to each point in the undistorted depth image, using (3)

$$X_{\text{IR}} = \frac{(x - O_{x\_\text{IR}})\, du(x,y)}{f_{x\_\text{IR}}},$$
$$Y_{\text{IR}} = \frac{(y - O_{y\_\text{IR}})\, du(x,y)}{f_{y\_\text{IR}}}, \tag{3}$$
$$Z_{\text{IR}} = du(x,y),$$

where $(X_{\text{IR}}, Y_{\text{IR}}, Z_{\text{IR}})$ 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, $(f_{x\_\text{IR}}, f_{y\_\text{IR}})$ is the focal length of the IR camera, $(O_{x\_\text{IR}}, O_{y\_\text{IR}})$ is the optical center of the IR camera, and $du(x, y)$ is the depth of a pixel in the undistorted depth image. Next, the color is assigned from the RGB image to each 3D point $P_{\text{IR}}(X_{\text{IR}}, Y_{\text{IR}}, Z_{\text{IR}})$. The color is mapped by transforming the 3D point P_{IR} into the color camera reference frame using the internal extrinsic camera parameters and then reprojecting that point on the image plane of the RGB camera using the intrinsic parameters to find the pixel location of the color in the undistorted color image using (4)

$$P_{\text{RGB}}(X_{\text{RGB}}, Y_{\text{RGB}}, Z_{\text{RGB}}) = R \cdot P_{\text{IR}} + T,$$
$$x = \left(\frac{X_{\text{RGB}} f_{x\_\text{RGB}}}{Z_{\text{RGB}}}\right) + O_{x\_\text{RGB}}, \tag{4}$$
$$y = \left(\frac{Y_{\text{RGB}} f_{y\_\text{RGB}}}{Z_{\text{RGB}}}\right) + O_{y\_\text{RGB}},$$

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 IR camera, and (x, y) is the pixel location of color information in the undistorted color image.

Figure 6(a) shows a mockup car door as imaged from the color camera; Figure 6(b) shows the colored depth information in the interval 0–2.5 m from the slightly different point of view of the IR camera, while keeping the Kinect sensor static with respect to the panel. The difference in position and orientation between the two cameras contained in the Kinect unit is accurately compensated by the estimated extrinsic parameters obtained from internal calibration.

4.2. External Calibration of Kinect Sensors with a Best-Fit Plane Method. The last set of parameters estimated in the calibration process is the extrinsic ones, which is the relative position and orientation between every pair of Kinect sensors. The external calibration is performed between pairs of IR cameras over the network of sensors because depth information is generated with respect to these cameras. The concept behind the proposed method, named here best-fit plane calibration method, is to determine, for every pair of sensors, the position and orientation of a fixed planar chessboard in real world coordinates. Knowing the orientation of the plane from two different points of view (i.e., two Kinect sensors), it is possible to estimate the relative orientation and position change between the sensors.

The procedure developed for external calibration involves positioning a standard planar chessboard target within the visible overlapping regions of any two Kinect sensors. Unlike most calibration techniques in the literature, in this method there is no need to move the checkerboard to image it from multiple views. On the contrary, a fixed target increases the performance of the method. The result is a rigid body transformation that best aligns the data collected by a pair of RGB-D sensors. Figure 7 depicts the proposed scanning system during the calibration process.

The proposed technique takes advantage of the rapid 3D measurement technology embedded in the sensor and provides registration accuracy within the range of the depth measurements resolution available with Kinect. An important advantage of this method is the fact that it is unnecessary to

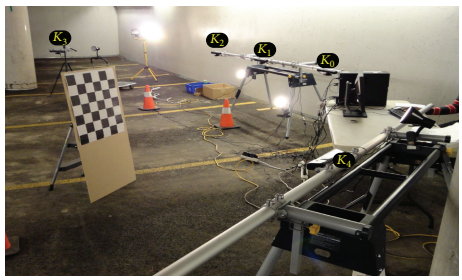
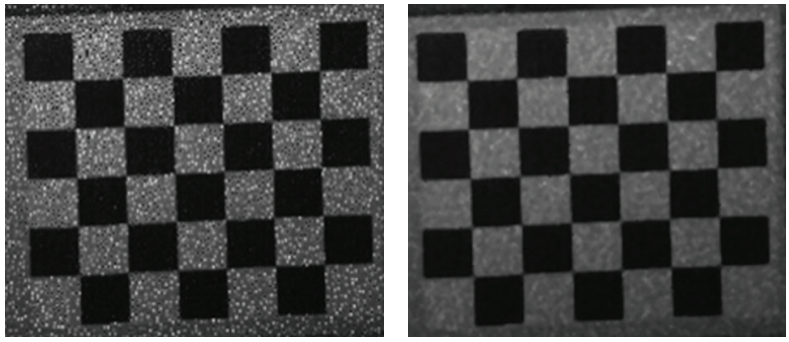


Figure 7: Placement of calibration target during calibration of Kinects K_0 and K_4.



(a) (b)

Figure 8: IR image of the checkerboard target for external calibration: (a) effect of the projected IR pattern, (b) filtered image using a median filter of size 3×3.

cover the Kinect infrared projector to perform this phase of the calibration, which facilitates manipulations when remotely dealing with the network of Kinect devices.

The method consists in finding a normal vector and the center of the checkerboard plane, which define the relative orientation and translation of the checkerboard plane. The first step is to compute the 3D coordinates of the corners on the checkerboard with respect to the IR camera frame, using (3). When the checkerboard target is positioned in front of a Kinect sensor, the IR projector pattern appears on the checkerboard target as shown in Figure 8(a). This pattern creates noise and makes it difficult to extract the exact corners. Since the noise is similar to salt and pepper noise, a median filter of size 3×3 provides a good reduction in the noise level without blurring the image, as shown in Figure 8(b).

Moreover, the extracted points are not entirely mapped over a single plane because of quantization effects in the Kinect depth sensor. Therefore, the corner points are used to estimate the three-dimensional plane (5) that minimizes the orthogonal distance between that plane and the set of 3D points. The equation of the plane then permits to estimate the orientation in 3D space of the target with respect to the IR camera

$$z = Ax + By + C. \tag{5}$$

Let the 3D coordinates of the n corners of the checkerboard target be $S_1(x_1, y_1, z_1), S_2(x_2, y_2, z_2), \ldots, S_n(x_n, y_n, z_n)$; then the systems of equations for solving the plane equation are $Ax_1+By_1+C = z_1, Ax_2+By_2+C = z_2, \ldots, Ax_n+By_n+C = z_n$. These equations can be formulated into a matrix problem

$$\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ \vdots & \vdots & \vdots \\ x_n & y_n & 1 \end{bmatrix} \begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix}. \tag{6}$$

This overdetermined system is solved for the values of A, B, and C with an orthogonal distance regression approach [21], which provides the best fit plane on those points. All the 3D points, S_n, are projected on the best fit plane, P_n. These points serve to define the center and the normal vector of the plane. However, projected points, P_n, do not represent the exact corners of the checkerboard. Therefore, the center of the plane cannot be defined only by the intersection of two lines passing close to the center. Figure 9(a) shows the set of possible pairs of symmetric corner points that generate lines passing close to the center.

The closest point to all intersections between these lines is selected as a center point O. Two points P_i and P_j are selected on the plane to define vectors $\overrightarrow{OP}_i$ and $\overrightarrow{OP}_j$. The normal to the plane is then defined by the cross product:

$$z = \frac{\overrightarrow{OP}_i \times \overrightarrow{OP}_j}{\left|\overrightarrow{OP}_i \times \overrightarrow{OP}_j\right|}. \tag{7}$$

This normal is the unit vector of the z-axis of the checkerboard frame with respect to the RGB-D sensor. The unit

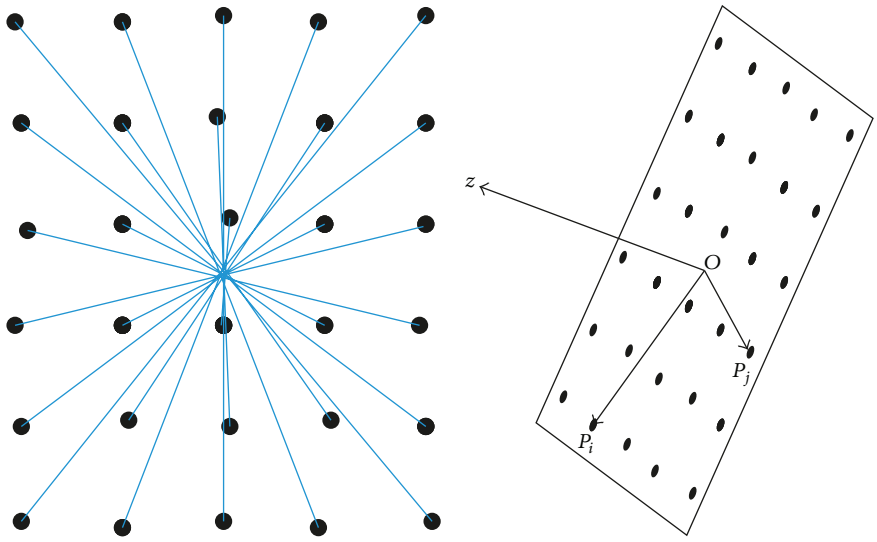


FIGURE 9: (a) Possible combination of lines passing through the center of the checkerboard, (b) the normal vector and the center of a checkerboard target.

vector of the y-axis of the checkerboard can be found by any two vertical points in the checkerboard. Let P_i and P_j be the two vertical points where P_i is the top end and P_j is the bottom end of a vertical line. N is the total number of possible combinations of vertical lines. The average unit directional vector can then be defined as

$$y = \frac{1}{N} \sum \frac{P_i - P_j}{\left| P_i - P_j \right|}. \tag{8}$$

This vector is the unit vector of the y-axis of the checkerboard frame with respect to the RGB-D sensor. The last unit vector for the x-axis can be found by a cross product, defined as

$$x = y \times z. \tag{9}$$

All the unit vectors of the coordinate frame of the checkerboard target can be combined to define the rotation matrix between the RGB-D sensor and the checkerboard frame as

$$R = \begin{bmatrix} x_x & y_x & z_x \\ x_y & y_y & z_y \\ x_z & y_z & z_z \end{bmatrix}. \tag{10}$$

The translation vector corresponds with the center of the checkerboard frame

$$T = \begin{bmatrix} O_x & O_y & O_z \end{bmatrix}. \tag{11}$$

R and T are the rotation and the translation matrices of the checkerboard frame with respect to the Kinect IR sensor. The position and orientation between two Kinect sensors can be determined by the following procedure. Let H_1 and H_2 be the homogenous transformations between Kinect 1 and the checkerboard and between Kinect 2 and the checkerboard, respectively, as shown in Figure 10. If the target remains

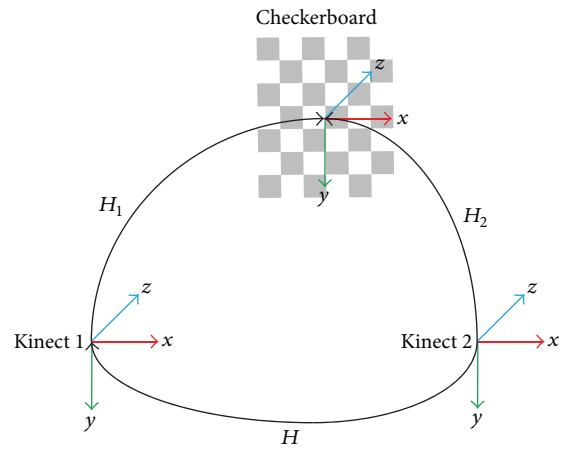


FIGURE 10: Extrinsic calibration of a pair of Kinect sensors.

fixed for both Kinect sensors, the geometrical transformation between the sensors is defined as follows:

$$H = H_2 H_1^{-1}, \tag{12}$$

where H is the homogenous transformation matrix from the Kinect 2 to the Kinect 1 sensor.

4.3. Complete Calibration of All Cameras in the Network of Kinect Sensors. The camera arrangement shown in Figure 4 includes overlapping regions between contiguous sensors marked in gray. During the calibration phase, the checkerboard target is successively placed within those areas for external calibration between every pair of neighbor Kinect IR sensors. Figure 7 shows the calibration target placed in the overlapping region between Kinect K_0 and K_4 during an experimental calibration procedure. External calibration is

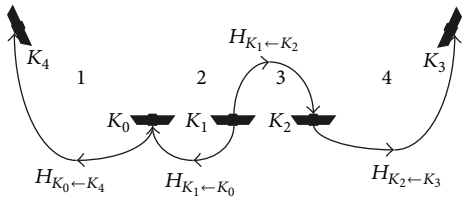


Figure 11: Calibration flow for the network of sensors.

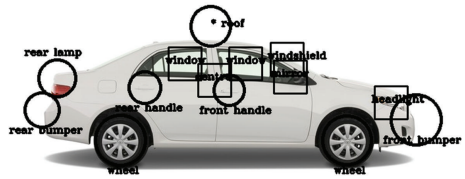


Figure 12: Automatic detection of parts of interest over a side view of a car.

performed in pairs using the best-fit plane calibration method detailed in Section 4.2 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 11 shows the calibration flow for the network of Kinect sensors.

Kinects K_0 and K_2 have a direct relationship with K_1, immediately defined by the extrinsic calibration parameters obtained in each case, but K_4 and K_3 need to be related to K_1 through an intermediate node, respectively, K_0 and K_2. The relations between (K_1, K_4) and (K_1, K_3) are given by the following equations:

$$H_{K_1 \longleftarrow K_4} = H_{K_1 \longleftarrow K_0} H_{K_0 \longleftarrow K_4},$$
$$H_{K_1 \longleftarrow K_3} = H_{K_1 \longleftarrow K_2} H_{K_2 \longleftarrow K_3}. \quad (13)$$

5. Automatic Detection of Characteristic Areas over a Vehicle

Once the network of Kinect sensors distributed around one side of a vehicle is available and entirely calibrated, the proposed system determines the location of the vehicle in the scene and subsequently the location of a set of significant vehicle components. The purpose of recognizing specific areas of interest over a large object such as a vehicle is to speed up the modeling process and also to facilitate the guidance of the robot arm that will eventually interact with the vehicle to perform either inspection or maintenance tasks. Acquiring the knowledge about the location of dominant features over the vehicle reduces the amount of time spent on scanning at higher resolution to accurately drive the manipulator by focusing the operation only over selected areas. It also allows the robot to rapidly determine where to operate, as it is very unlikely that the robotic operation will be required over the entire vehicle for most typical inspection or maintenance tasks.

To achieve efficient and reliable detection and localization of characteristic areas over a vehicle, a visual detector of vehicle parts (VDVP) was previously introduced in [22] to operate on an image depicting a complete view of one side of a vehicle. The VDVP receives as an input a color image of a lateral view of the vehicle to determine the location of up to 14 vehicle parts. The method works with images of different types of vehicles such as 4-door sedan, 2-door sedan, 3-door hatchback, 5-door hatchback, SUV and pickup-trucks. Figure 12 illustrates the result of applying the method over a test image. Round areas indicate features detected by the classifiers; square regions mean that the locations of the features were inferred based on other known features. The VDVP method is revisited and integrated in this work.

The original method required at least a full lateral view of the vehicle which was provided by a standard color camera located far enough from the vehicle to image its entire length. To better integrate and to increase the compactness of the acquisition setup, determining the location of the car and the bodywork panels of the vehicle should be achieved by using the color information provided by the Kinect sensors. However, none of the Kinect sensors can collect a full view alone due to the limited horizontal field of view and depth of field of Kinect sensor technology. Therefore, it becomes necessary to merge the color images collected in a collaborative manner by sensors K_0, K_1, K_2. For that purpose, the assumption is made that the image planes of the group of Kinect sensors are parallel and they all are contained in a larger plane. Thus, in order to accurately merge the color information, only the translation vectors defining the relative position of the color cameras with respect to the central Kinect's color camera are required. These parameters were obtained during the calibration procedure described in Section 4 and can be computed by combining the internal and external extrinsic calibration parameters, as detailed in Section 4.3. The resulting composite image is used to determine the segments in each color image that correspond to both the vehicle and the parts of interest. Here the original VDVP method is extended such that the detection process for each part of interest is performed using individual images from each sensor, each representing a section of the vehicle rather than on the segmented panorama.

To determine the location of a car in the side view, the VDVP method relies on the successful detection of the car wheels [22]. To tackle this problem, the algorithm makes a decision based on the result of two techniques that estimate the position of the wheels: the Hough transform algorithm and a classifier based on Haar-like features trained for detecting wheels [16]. Then, the car is enclosed in a bounding box and segmented from the background. To set the dimension of the bounding box, a statistical analysis of the location of significant car features over a training set of images (t_s) was conducted. The training set represents the 67% of a collection containing one hundred and ten color images. The remaining 33% of the images were used for evaluation purposes. Figure 13 shows the regions where features of interest are expected to be located. Each region of interest corresponds, from left to right, respectively, to *rear bumper, rear lamp, rear wheel, rear handle, rear window, roof, center, front handle, front window, windshield, lateral mirror, front wheel, head light, and front bumper.*

Figure 13: Regions containing the location of parts of interest for lateral views of vehicles, as a result of training over a representative collection of images of different types of automobiles.

Having recovered the position of the wheels in the scene, for parts of interest detection purposes, a polar coordinate system is established. The origin is the center of the rear wheel. Directions X and Y are at right angles to each other. The direction of the X axis matches with the directed line that goes from the origin (center of rear wheel) to the center of the front wheel.

Once the location of the wheels is known, the next step consists in determining the location of all other remaining features of interest, C_i, on the bodywork. Any feature of interest, C_i, characterizing a vehicle is defined in the polar coordinate system, as follows:

$$C_i = \{(r, \theta, \delta) : r, \delta \in R, 0 \leq \theta \leq \pi\}, \tag{14}$$

where r and θ define the center of a feature of interest to be located, in polar coordinates, and δ is the minimum radius of a circumference enclosing the region for a given part of interest. Let R_c be the region where a feature, C, is expected to be located. R_c *is a 2-tuple* (f_r, f_θ) that can be represented by probability distribution functions over the polar map superimposed over the lateral image of the vehicle and defined as

$$\begin{aligned} f_r\left(r; \mu_r, \sigma_r^2\right) &= \frac{1}{\sigma_r\sqrt{2\pi}} e^{-((r-\mu_r)^2/2\sigma_r^2)}, \\ f_\theta\left(\theta; \mu_\theta, \sigma_\theta^2\right) &= \frac{1}{\sigma_\theta\sqrt{2\pi}} e^{-((\theta-\mu_\theta)^2/2\sigma_\theta^2)}. \end{aligned} \tag{15}$$

The radial and angular standard deviations (σ_r and σ_θ) and expectations (μ_r and μ_θ) in (15) are calculated experimentally from the training set, t_s. Consequently, these PDF functions define a probable searching area for each part of interest considered.

To achieve rotation, scale and translation invariance in the definition of the search zones, the direction of the vector pointing toward the center of the front wheel is used as the X axis, all the vectors pointing toward features of interest were normalized with respect to the length between the wheels' centers, and the origin of the coordinate system corresponds to the position of the rear wheel's center as it is shown in Figure 14(b). Up to this point a set of regions for each part of interest and for each image, in the segmented panorama, was defined. Next, for each car feature, C_i, to be detected, a search area, R_{c_i}, is defined. Then, a set of classifiers trained for detecting each feature of interest, C_i, is used. The detection method is constructed as a cascade of boosted classifiers based on Haar-like features. Classifiers were trained with the set of images, t_s. Each detection performed by a classifier is ranked using the corresponding PDF functions (f_r, f_θ). False detections are discarded rapidly using this method as well.

5.1. Determining the Position of Missing Vehicle Parts. Vehicle panels are located using vehicle features on the bodywork. Figure 15 shows the classification rate for each car part. The classifiers were evaluated using the testing set that, as mentioned previously, contains the 33% of a collection of color images of vehicles. Twelve features are detected by the learning algorithms; the locations of the remaining features (windows) as well as the location of those characteristics that were not successfully detected by the classifiers are inferred as follows.

Let M be a geometrical model for a vehicle defined as follows: $M = \{C_1, C_2 \ldots C_n\}$. That is, M defines the spatial relationship among the vehicle features of interest in an image by containing the positions of each car part.

Let $M_c = \{C_1, C_2 \ldots C_{n-k}\}$ a geometrical model constructed using the vehicle parts of interest successfully detected with the method proposed in the previous section, k being the number of missing features. Let G be the set of geometrical models for each image in the training set, t_s. A similarity function, L, which measures how adequate M_c and M are, can be defined as follows:

$$L\left(M_c, M\right) = \sum_{i=1}^{k} S\left(c_i\right) \cdot F\left(c_i\right), \tag{16}$$

where $S(c_i)$ is the probability of the successful detection of the classifier for a particular feature, c_i. This probabilistic distribution was determined experimentally for each car part to be detected and is reported for dominant parts of interest in [22]. $F(c_i)$ defines the probability that a detected feature, c_i, was found in the right place considering the model, M, and the probability distribution associated with c_i. Therefore, for each feature, c_i, in M_c, the similarity function, L, is calculated using models in G. This way the most likely position of c_i is determined by choosing the lowest value for L. The partial model, M_c, is upgraded with the new location of c_i. This process is repeated until the full set of features is known. Finally, the model, M_c, not only provides accurate information about the location of the parts of interest but information about the type of car, size, location, and orientation of the vehicle since each model M in G beyond containing the spatial relationship among the vehicle features of interest also contains semantic information about the type of vehicle. Figure 16 shows some results obtained after applying the proposed technique for part detection in color images collected from three Kinect sensors over different lateral sections of a vehicle.

6. 3D Reconstruction of Vehicle Lateral Panels

The point clouds collected by the set of Kinect sensors are first aligned using the external calibration parameters previously calculated. Then, a segmentation procedure is applied to separate points contained within discovered regions of interest, using the VDVP method of Section 5, from the whole point cloud. For segmentation, the locations of the detected parts of

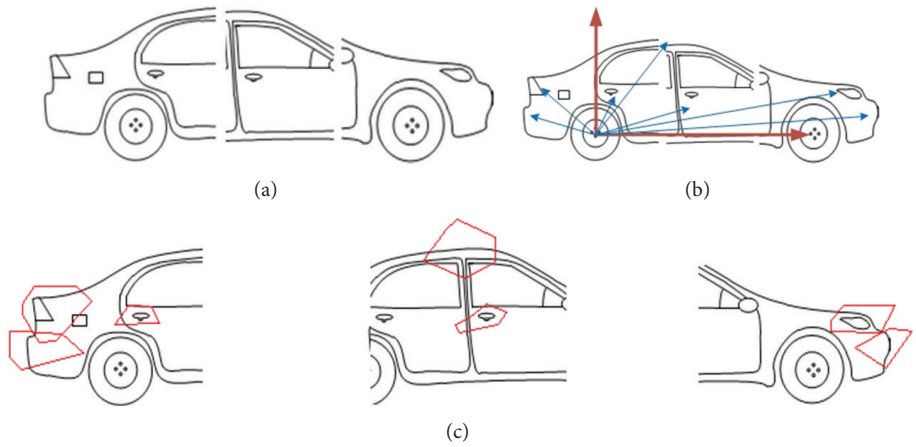


FIGURE 14: (a) Unrefined segmented panorama, (b) polar coordinate system centered on the rear wheel and the centers of each search zone, and (c) search zones R_c for each image.

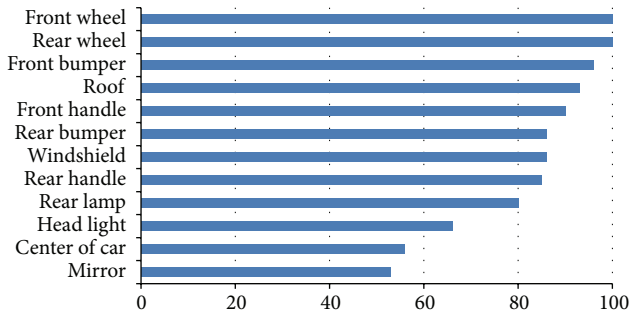


FIGURE 15: Classification rate according to car parts in percentage.

interest in color images are mapped into the 3D point cloud. The coordinates of the rear bumper, front bumper, roof, and wheels provide the width and height of a 3D bounding box enclosing the desired points. The bonding box depth is defined by adding and subtracting a fixed parameter (30 cm) to the average depth of the parts. This parameter was established experimentally, as folds and pleats of the surface of automotive vehicles usually do not go beyond this threshold. Next, a triangle mesh is built over the respective groups of points that correspond to each region of interest detected over the surface. This process results in a group of local 3D colored models that represent the shape and visual appearance of the surface of the vehicle in the specific regions of interest. A mesh is a topological structure typically constructed from unorganized points. The Delaunay triangulation technique [23] is commonly used for this purpose but tends to be computationally expensive. Nevertheless, the Kinect sensors provide depth readings in a structured way (rectangular lattice). Advantage is taken of this structured information to perform the triangulation process efficiently using the following technique.

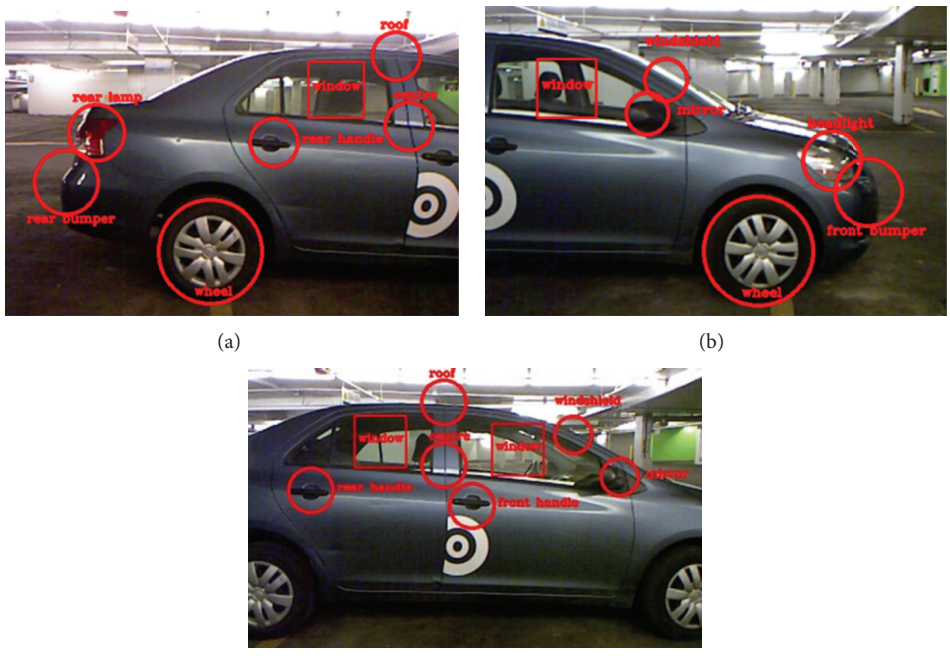
Let TR_m be the triangle mesh to be calculated for a given point cloud P_L. The latter being a rectangular lattice of 3D points under the constraint that Kinect sensors are used. Let h be a variable threshold that defines the surface continuity and let pe, ps, pw, pn be the four neighbors of a point pi defined as shown in Figure 17. The parameter h increases in accordance with the distance between the sensor and the surface due to the fact that the error of the sensor increases along the z-axis.

Now, let us assume that pi, pe, ps, pw, and pn are indices of the point cloud vector and that a 3D point v_i can be accessed in this way: $v_i \leftarrow P_L$ (index). If there is a hole in P_L, the z coordinates of the missing points are set to ∞. Given this structure, the triangle mesh generation is accelerated as follows.

For each point i in P_L, two candidate triangles are evaluated individually according to the continuity criteria of their vertices. The criterion for the inclusion or rejection of a triangle consists in comparing the threshold h, mentioned previously, to the difference of the z-coordinates between pairs of points. If the check is passed successfully, the three-sided polygon is added to the mesh. The triangles are defined as follows: $T_{m_1}[j] \leftarrow$ triangle (v_1, v_2, v_3) and $T_{m_2}[j] \leftarrow$ triangle (v_1, v_4, v_5), where $v_1 = P_L(i)$, $v_2 = P_L(e)$, $v_3 = P_L(s)$, $v_4 = P_L(w)$, and $v_5 = P_L(n)$. By using this technique, the triangulation process is performed rapidly. The downside of the method is the fact that it tends to leave small holes if there are missing points in the point cloud.

7. Experimental Evaluation

This section presents a series of experiments aimed to evaluate the proposed sensing framework. The first experiment assesses the tolerance of the sensors when they capture irregular shapes of car body parts over simulated uneven surfaces such as that in semicontrolled environments. The framework provides point clouds of the different sections of the vehicle; the information is aligned and it is used to create a triangle mesh of the surface. Thus, the next experiment evaluates the performance of the proposed mesh generation algorithm. This section closes with a qualitative evaluation of full 3D reconstructions of lateral panels of real vehicles and large objects.



(a) (b)

(c)

FIGURE 16: Automatic detection of fourteen parts of interest over three different RGB images which correspond to different sections of a vehicle: (a) view from sensor K_0, (b) sensor K_2, and (c) sensor K_1.

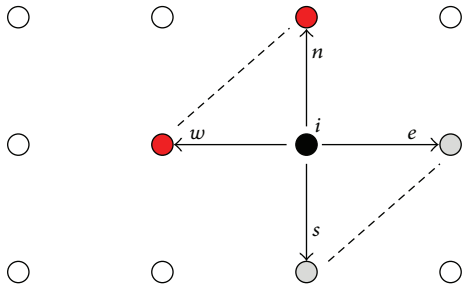


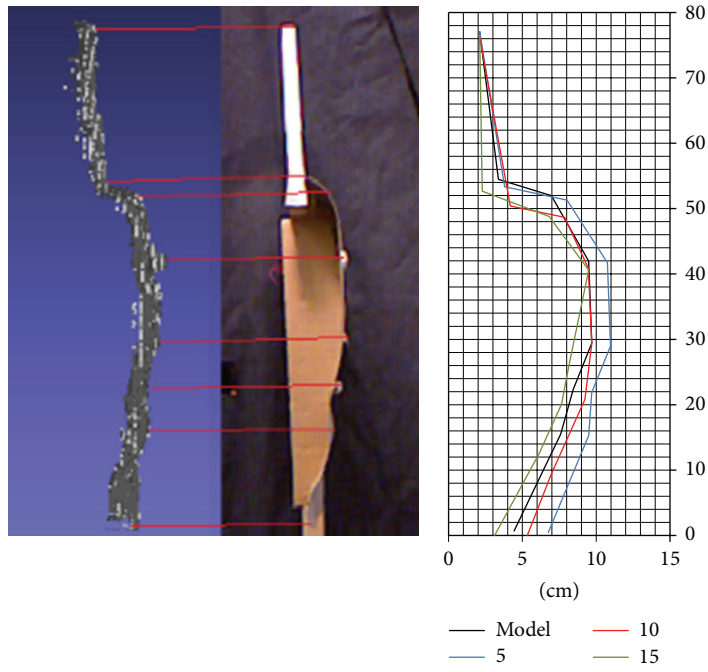
FIGURE 17: 2D representation of a 3D point cloud; the z-coordinate of each point was removed. The solid dot is a vertex i shared by the triangles (i, e, s) and (i, w, n).

In order to evaluate the capability of the system to reconstruct irregular surfaces, a model of a car door was used, as shown in Figure 6(a) and was imaged with a single Kinect sensor. The model was designed for simulating the irregular curves of an actual vehicle surface as well as folds and pleats, as shown in Figure 18(a). A set of salient points over the surface of the door was selected to evaluate the accuracy of the method. Using these points, a silhouette of the panel is built. Figure 18(a) shows the salient points in both the door model and its reconstruction. Five points of the outline represent tight curves over the surface. Additionally, three small bumps were added over the smooth part of the surface to be used as reference points. Figure 18(b) presents the results of constructing the silhouettes for the door model using different point clouds. The data were collected with the sensor tilted at angles of 5, 10, and 15 degrees with respect to the horizontal level. The sensor was positioned at 30 cm above the bottom of the door and 1.5 m from the closest point of the door. The uppermost points of the outlines were selected as a common point for the alignment of the silhouettes.

In Figure 18(b), the black line represents the actual shape of the model. Color lines represent results from the experiments under three different inclinations of the sensor with respect to the object. The error along the y and z coordinates, for most of the points, remains in the interval $(0, \pm 1.8)$ cm with respect to the actual door model which proves that the scanning system achieves a good degree of tolerance to irregular leveling and alignment of the sensors. The image shown in Figure 19 is the 3D reconstruction and 3D mesh of the mock-up door. For visualization purposes in this paper, the Quadric Clustering decimation algorithm [24] was applied over the triangle mesh to reduce the number of triangles.

The next experiment was conducted to measure the performance of the proposed algorithm for triangle mesh generation. The results are compared, in Table 3, with those obtained by using the Delaunay algorithm provided by the Visualization Toolkit library (VTK) [25]. Sensors K_0, K_1, K_2 were used to create three individual triangle meshes for their corresponding bodywork region. The results shown in Table 3 were obtained after running the algorithms 10 times over each point cloud. The difference in the results can be explained by the fact that the Delaunay algorithm is designed to work with nonstructured 3D points while the proposed technique from Section 6 takes advantage of the rectangular lattice of 3D points produced by the Kinect sensors.

Lastly, experiments were conducted using the proposed scanning system and real vehicles. In this test, a full reconstruction of the lateral panels of different vehicles was achieved to evaluate the capability of Kinect sensors to



(a) (b)

Figure 18: (a) Profile of a car door model over which a set of salient points were selected to construct the outline of the surface for evaluation purposes, and (b) silhouette reconstruction with the sensor tilted at different angles.

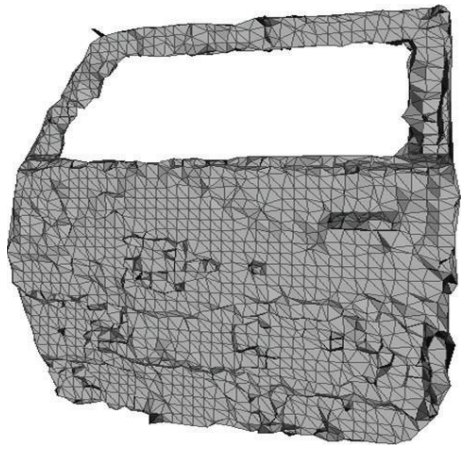


Figure 19: Triangle mesh of a mockup door model.

Table 3: Comparison of algorithms performance for triangle mesh generation.

Vehicle panel	Average number of vertices	Delaunay algorithm, average time	Proposed triangle mesh generation algorithm, average time
Front	174687	3.09 sec	0.086 sec
Central	219286	3.65 sec	0.171 sec
Back	217308	3.63 sec	0.138 sec

perform in semiconstrained environments and over large objects exhibiting complex surface reflectance characteristics.

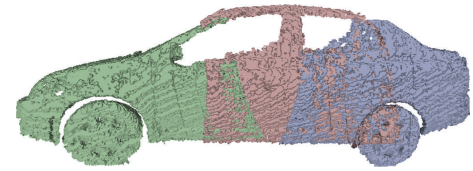


Figure 20: Point cloud registration of three different lateral sections of a 4-door sedan vehicle.

The first acquisition was performed in a semioutdoor parking garage over day time. Natural light was present in the scene via peripheral openings in the garage, while the sensors were protected from direct sunlight by means of the concrete ceiling. Figure 20 shows a sample of 3D reconstruction results obtained with the proposed methodology and a network of three Kinect sensors for the vehicle depicted in Figure 1(b). This acquisition was performed over winter season in Canada resulting in the vehicle's side panels being covered with dirt and salt deposits from the road conditions, which created various shades of green paint, gray dirty areas, and specular reflection spots from the overhead lighting present in the installation. The raw information collected by each Kinect can be distinguished by its color for the front, central, and back panels. Figure 21 shows the complete 3D reconstruction results after registration and fusion based on the calibration phase performed with the proposed methodology. For visualization purposes a decimation algorithm was applied over the mesh such that triangles are large enough to be visible. This also impacts on the apparent smoothness of the displayed model. The 3D reconstruction is provided in the Stanford

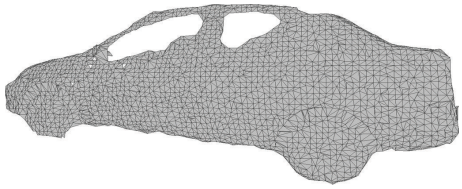


Figure 21: Resulting mesh for a Toyota Echo vehicle.

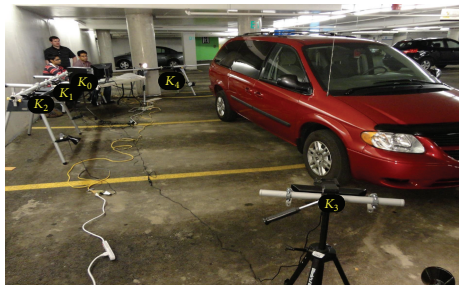


Figure 22: Capturing 3D data over a vehicle with the network of Kinect sensors.

Triangle Format [26], which is a convenient and standard way to represent the information of 3D meshes, and it can be used easily by a robot to plan its trajectory to approach the surface.

For the next experiment the acquisition was performed in an indoor parking garage. Figure 22 shows a vehicle standing in front of the experimental setup with 5 Kinect sensors for the rapid 3D modeling stage that will drive the robotic inspection. The scene was illuminated by incandescent lamps and a halogen lamp.

The reconstruction for two different types of vehicles is shown in Figures 23 and 24. In this case the texture captured by the Kinects' RGB cameras is added to the reconstructed scene, which provides a better appreciation of the reconstruction.

The windshield, lateral windows, and part of headlamp and rear lamp are missing in the depth maps because the IR energy generated by the Kinect devices passes through the transparent surfaces or is deflected in other directions. However, the rear window of the minivan, which is made of tinted glass, is partially captured. All of the main areas of the vehicles body and wheels, including dark rubber tires, are accurately reconstructed, and sections of the model acquired from the five viewpoints are correctly aligned, even over narrow roof supporting beams and highly curved bumpers areas.

Table 4 presents a comparison between the characteristics of the reconstructed vehicle and their actual dimensions. The Kinect depth quantization introduces scaling errors of about 1 cm in height and width and a depth error of about 2.5 cm at 3 m distance. Each sensor covers the full height of the vehicle, and the average error on height is under 1%. The estimation of the length of the vehicle and the wheel base (i.e., the distance between the centers of the front and back wheels) involves all the calibration parameters estimated for the network of Kinect sensors. The error on the length is under 2.5%, which is relatively minor given the medium quality of data provided by

Figure 23: Six different views of a minivan vehicle 3D reconstruction.

Figure 24: Six different views of a semicompact vehicle 3D reconstruction.

Table 4: Reconstruction compared with ground truth.

	Height	Length	Wheel base
Car			
Actual (mm)	1460	4300	2550
Model (mm)	1471	4391	2603
Error (%)	**0.75**	**2.11**	**2.07**
Van			
Actual (mm)	1748	5093	3030
Model (mm)	1764	5206	3101
Error (%)	**0.91**	**2.21**	**2.34**

Kinect at a depth of 3 m and in proportion to the large working volume. For further assessment of the algorithm, an ICP algorithm [27] was applied on the point clouds, but it did not significantly improve the registration over what was achieved with the estimated calibration parameters. This confirms the accuracy of the initial calibration described in Section 4.

Finally, Figure 25 shows the reconstruction of other models of vehicles along with that of some garbage bins, acquired with the exact same setup, to evaluate the generalization capabilities of the proposed calibrated RGB-D acquisition framework. A wide range of vehicles was covered during experiments, in terms of colors and size. The white color vehicle appears more integrally than the vehicles with dark gray color, where missing depth data are noticed over the front part on the right of the vehicles where the density of points in the acquisition varies to a greater extent given the significant

Figure 25: Reconstruction of various vehicles and garbage bins.

change of alignment between Kinects K_2 and K_3. The dark green garbage bins are also correctly reconstructed with proper alignment between the piecewise RGB-D models.

The algorithms presented in this paper were developed in C++ and run on a computer with an Intel core i7 CPU and Windows 7. The average time that the proposed method takes to reconstruct the 3D surface shape captured from each viewpoint for a regular vehicle is 4.0 sec. With regards to the acquisition time, 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 vehicle are completed within 30 seconds. It is worth to say that most of that time is consumed by the subsystem for the visual detection of parts of interest. The calibration process is performed off-line, and the triangulation algorithm is run as the sensors are collecting the depth information.

The automated selection of regions of interest detailed in Section 5 allows for rapid extraction of subsets of the generated 3D model over which further processing can be performed, including higher resolution scanning, if required, over only limited but strategically selected surfaces in order to drive a robotic operation with higher precision over those regions. This proves an efficient strategy given that very high resolution acquisition and 3D modeling over an entire object of the size of a vehicle would be prohibitive in time and resources.

8. Conclusions

In this work, a rapid acquisition and reconstruction methodology for automated 3D modeling of large objects such as automotive vehicles is presented. The approach builds on a network of fast Kinect sensors distributed around the object to collect color and depth information over lateral panels of vehicles. The 3D modeling results are meant to provide a robotic arm with sufficiently accurate spatial information about the bodywork of a vehicle and the 3D location of up to fourteen features of interest over the surface such that it can interact with the automobile panels for various inspection or maintenance tasks.

This technology opens the door to a great number of real-time 3D reconstruction applications using low-cost RGB-D sensors. The main contributions of this work provide a reliable methodology to integrate multiple color and depth streams of data in the task of 3D reconstruction of large objects. For that purpose, an efficient method for complete and accurate calibration of all intrinsic and extrinsic parameters of RGB-D sensor units specifically tailored to the Kinect sensors technology is presented. Furthermore, an approach for automated detection and recognition of critical regions of interest over vehicles from a mosaic of Kinect's color images is detailed. Finally, an accelerated triangular mesh generation algorithm is designed that takes advantage of the intrinsic structure of range data provided by Kinect sensors to further speed up the 3D model generation. The entire framework is experimentally validated under several operating conditions, including a laboratory environment, and semicontrolled parking garages where acquisition and 3D reconstruction are performed over objects of various sizes, including large vehicles. The results demonstrate the validity, accuracy, and rapidity of the use of Kinect's RGB-D sensors, in the context of robotic guidance. The addition of extra sensors to achieve full 360 degree coverage of a vehicle represents the next step of this investigation which will further extend the current capabilities.

Conflict of Interests

The authors declare that there is no conflict of interests with any company or organization regarding the material discussed in this paper.

Acknowledgments

The authors acknowledge the support from the Natural Sciences and Engineering Research Council of Canada to this research via the Strategic Project Grant no. STPGP 381229-09, as well as the collaboration of Scintrex Trace Corp.

References

[1] J. Zhou, J. Tong, L. Liu, Z. Pan, and H. Yan, "Scanning 3D full human bodies using kinects," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 4, pp. 643–650, 2012.

[2] A. Maimone and H. Fuchs, "Encumbrance-free telepresence system with real-time 3D capture and display using commodity depth cameras," in *Proceedings of the 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR '11)*, pp. 137–146, October 2011.

[3] P. J. Noonan, T. F. Cootes, W. A. Hallett, and R. Hinz, "The design and initial calibration of an optical tracking system using the microsoft kinect," in *Proceedings of the IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC '11)*, pp. 3614–3617, October 2011.

[4] P. Rakprayoon, M. Ruchanurucks, and A. Coundoul, "Kinect-based obstacle detection for manipulator," in *Proceedings of the IEEEE/SICE International Symposium on System Integration (SII '11)*, pp. 68–73, 2011.

[5] K. Berger, K. Ruhl, M. Albers et al., "The capturing of turbulent gas flows using multiple kinects," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV '11)*, pp. 1108–1113, Barcelona, Spain, November 2011.

[6] J. Smisek, M. Jancosek, and T. Pajdla, "3D with kinect," in *Proceedings of the IEEE International Conference on Computer*

Vision Workshops (ICCV '11), pp. 1154–1160, Barcelona, Spain, November 2011.

[7] C.-S. Park, S.-W. Kim, D. Kim, and S.-R. Oh, "Comparison of plane extraction performance using laser scanner and kinect," in *Proceedings of the 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI '11)*, pp. 153–155, Seoul, Korea, November 2011.

[8] N. Burrus, "Demo software to visualize, calibrate and process kinect cameras output," 2012, http://nicolas.burrus.name/index.php/Research/KinectRgbDemoV6.

[9] M. Gaffney, "Kinect/3D scanner calibration pattern," 2011, http://www.thingiverse.com/thing:7793.

[10] K. Berger, K. Ruhl, Y. Schroeder, C. Brummer, A. Scholz, and M. Magnor, "Markerless motion capture using multiple color-depth sensors," in *Proceedings of the Vision, Modeling and Visualization*, pp. 317–324, 2011.

[11] K. Khoshelham, "Accuracy analysis of kinect depth data," in *Proceedings of the ISPRS Workshop on Laser Scanning*, pp. 1437–1454, 2011.

[12] S. Matyunin, D. Vatolin, Y. Berdnikov, and M. Smirnov, "Temporal filtering for depth maps generated by kinect depth camera," in *Proceedings of the 5th 3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON '11)*, pp. 1–4, May 2011.

[13] S.-M. Kim, Y.-C. Lee, and S.-C. Lee, "Vision based automatic inspection system for nuts welded on the support hinge," in *Proceedings of the SICE-ICASE International Joint Conference*, pp. 1508–1512, October 2006.

[14] S. Agarwal, A. Awan, and D. Roth, "Learning to detect objects in images via a sparse, part-based representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1475–1490, 2004.

[15] A. Kiryakov, B. Popov, I. Terziev, D. Manov, and D. Ognyanoff, "Semantic annotation, indexing, and retrieval," *Journal of Web Semantics*, vol. 2, no. 1, pp. 49–79, 2004.

[16] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01)*, pp. I511–I518, Kauai, Hawaii, USA, December 2001.

[17] Y.-F. Fung, H. Lee, and M. F. Ercan, "Image processing application in toll collection," *IAENG International Journal of Computer Science*, vol. 32, pp. 473–478, 2006.

[18] M. M. Trivedi, T. Gandhi, and J. McCall, "Looking-in and looking-out of a vehicle: computer-vision-based enhanced vehicle safety," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 1, pp. 108–120, 2007.

[19] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[20] Open Source Computer Vision Library, http://opencv.willowgarage.com/wiki/.

[21] Real-time Computer Graphics and Physics, Mathematics, Geometry, Numerical Analysis, and Image Analysis, "Geometrictools," http://www.geometrictools.com/LibMathematics/Approximation/Approximation.html.

[22] A. Chávez-Aragón, R. Laganière, and P. Payeur, "Vision-based detection and labelling of multiple vehicle parts," in *Proceedings of the IEEE International Conference on Intelligent Transportation Systems*, pp. 1273–1278, Washington, DC, USA, 2011.

[23] M. De Berg, O. Cheong, M. Van Kreveld, and M. Overmars, "Delaunay triangulations: height interpolation," in *Computational Geometry: Algorithms and Applications*, pp. 191–218, Springer, 3rd edition, 2008.

[24] P. Lindstrom, "Out-of-core simplification of large polygonal models," in *Proceedigs of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '00)*, pp. 259–262, July 2000.

[25] The Visualization Toolkit, VTK, http://www.vtk.org/.

[26] Stanford Triangle Forma, PLY, http://www.cs.virginia.edu/~gfx/Courses/2001/Advanced.spring.01/plylib/Ply.txt.

[27] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, February 1992.