2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Automated Data Processing for a Rapid 3D Surface Inspection System

Quan Shi, Dr. Ning Xi
Electrical and Computer Engineering
Michigan State University
East Lansing, Michigan, 48824
Email: shiquan@egr.msu.edu, xin@egr.msu.edu

***Abstract*— For 3D dimensional inspection systems, point clouds measured on each viewpoint need to be processed for quality evaluation. Three steps are usually included in this process: filtering, registration, and error map generation. For quality control, small defects like dints and dents have to be kept in the point cloud. Therefore, a filtering algorithm is required to automatically remove outliers and keep dints/dents. Many filtering algorithm smooth the point cloud for better display, however, since the measured point cloud is used to represent the shape of the part, modification of any point's coordinates is not allowed because that will modify the error map. A point cloud filtering algorithm is developed using a link clustering algorithm to identify and remove outliers. Point cloud filtering is especially important in an iterative closest point (ICP)-based robot hand-eye calibration method because outliers will bring calibration errors into the calculated transformation matrix. With this technique, the cleaned point clouds can be directly transformed to a world frame for registration. This registration method has two advantages compared to feature-based registration methods: 1) the entire inspection process can be automatically executed, 2) avoid holes in point clouds caused by artificial markers. For error map generation, a point-to-plane distance is used in this paper which calculates the distance of a point to its closest triangle. The introduced automated inspection system had been implemented on a PUMA robot system. Experimental results are described in this paper.**

***Index Terms*— Point cloud registration, link clustering, automatic dimensional inspection system.**

I. INTRODUCTION

A robot-aided 3D dimensional inspection system had been developed for automatic quality evaluation of a part surface [1][2]. This system can automatically move an area sensor above a part and generate a point cloud on each viewpoint. However, the data processing for quality control, including point cloud registration and error map generation, often requires many manual operations that could not be automatically implemented. This paper introduces our methods of point cloud registration and error map generation. The methods are developed toward a fully automatic inspection process. After 3D shape measurement, each individual point cloud can be directly transformed into a common frame for quality verification.

A previous work had been developed for point cloud registration by robot kinematics [3], in which it was pointed out that iterative closest point (ICP)-based algorithms [4] may not fit inspection applications because the transformation matrix for registration is estimated in a sense that the total shape error is "minimized", which cannot be applied to industrial quality control. In another aspect, for a part surface that artificial markers are not allowed to be put on, the feature-based registration algorithms may not be able to execute. Those problems leave the transformation technique as the practical way for stitching point clouds together.

To automatically register point cloud together, a new ICP-based robot hand-eye calibration strategy had been proposed in [3]. However, two problems still need to be solved, 1) the accuracy of robot kinematics can hardly meet the requirements for industrial inspection; 2) the measured point clouds have to be cleaned for robot hand-eye calibration.

Robot link errors may cause position and orientation errors on robot tool center point (TCP), this error systematically exists because of the mechanical offsets of each robot joints. Calibration of each robot link can only reduce the error. However, since the mechanical offsets of each link is fixed, it is possible for an industrial robot to repeat the positioning "error". Which means, the robot can still be used as long as its repeatability satisfies the motion accuracy. Currently, repeatability of industrial robots can reach to several μms, a high level positioning system, such as a coordinate measurement machine (CMM), will have better performance than industrial robots. Therefore, transformation technique is a possible solution for point cloud registration. Hence, as long as transformation matrices from each viewpoint to this predefined world frame are calibrated, point clouds measured on those viewpoints can be matched together and registered to its CAD model in a world frame.

Robot hand-eye calibration [6] is often used to determine the transformation matrix between the robot end-effector and a vision sensor. In the developed robot-aided dimension inspection system, the matrix of robot hand-eye relationship is also required. Previously, we proposed an ICP-based hand-eye calibration technique, which derives a transformation matrix by iteratively matching two point sets. In this paper, we use the similar technique to determine transformation matrices from each viewpoint to the world frame.

In the practice of the developed ICP-based calibration method, we have to first clean the measured point cloud. As to the point cloud filtering, A smooth method is developed in [7] and an enhanced vector quantization method is proposed in [8]. For automatic surface inspection, coordinates of the measured points cannot be modified because the surface defects may not be revealed after a point cloud is smoothed. Also, since point clouds already contain a number of data point much larger than required by processing, algorithm efficiency is also required for real time computation. In this paper, we introduce a link clustering algorithm that is developed for outlier classification and cancellation from point clouds. The advantage of the proposed method is that 1) the size of outliers is controllable; 2) this algorithm will not modify any data point. Therefore, small defects such as dints/dents can be visualized, which is required in many surface inspection applications.

A new point cloud registration method will be introduced first in section II, followed by the link clustering method. Section IV introduces the details of an error map generation method. Experimental results are provided and conclusions are summarized at the end.

II. Registration of point cloud registration to the world frame of an inspection system

A previous work [3] had introduced the transformation techniques that use robot base frame as the common frame for point cloud registration. The measured surface data, locally in the frame of each viewpoint, can be transformed to robot base frame after the all robot links and hand-eye relationship are well-calibrated. However, calibration of robot links is not an easy task, which usually requires professional devices and operators. Usually, it was done by robot manufacturer. In applications of an industrial robot, robot repeatability is more emphasized than robot accuracy.

Robot accuracy specifies the uncertainty range that tells how the robot tool center point (TCP) is away from its programmed point. Robot repeatability shows an uncertainty range how a robot repeats itself to a target point. The range of repeatability is usually smaller than the the range of motion accuracy. Therefore, it is more accurate for a robot to be moved to a taught point rather than a programmed point. Then, to obtain a better registration result, point clouds can be transformed to a world frame instead of robot base frame because the relationship between each viewpoint to the world frame can be calibrated before online measurements.

As shown in Figure 1, W is the world coordinate frame, which is predefined in the measurement workstation. The robot will move the area sensor to predetermined viewpoints, represented by $V1$ and $V2$. $C1$ and $C2$ are the frames of the area sensor, in which a point cloud will be locally created. T_{V1}^{W} and T_{V2}^{W} are transformations from sensor frame to the world frame, which are required for point cloud registration. To determine T_{vi}^{W} for viewpoint i, T_{Ci}^{Pi} and T_{Pi}^{W} are required:

$$T_{Vi}^{W} = T_{Ci}^{Pi} \cdot T_{Pi}^{W}, i = 1, 2, 3... \tag{1}$$

Some gauges with standard shape are required to determine T_{Ci}^{Pi} and T_{Pi}^{W}. T_{Pi}^{W} can be easily determined by setting up

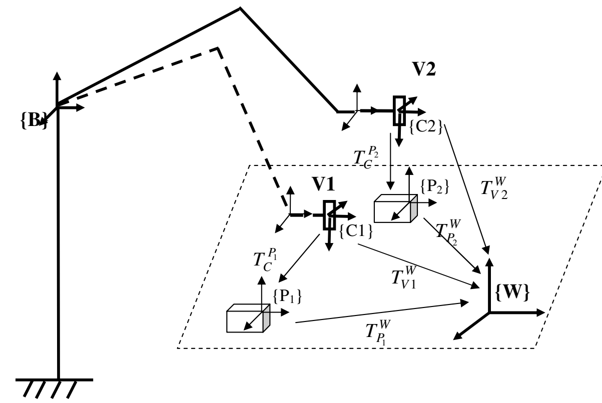


Fig. 1. Coordinate transformations in a robotic 3D shape measurement system

gauges at pre-calibrated locations in the inspection workstation. Coordinates of control points on the surface of gauges can also be determined. Therefore, by using an ICP-based shape fitting method, T_{Ci}^{Pi} can be derived as well.

$$P_C = \{p_i(x_i, y_i, z_i) | i = 1, 2...m\} \tag{2}$$

$$S_M = \{G_i | G_i =< (X_{i1}, Y_{i1}, Z_{i1}), (X_{i2}, Y_{i2}, Z_{i2}), (X_{i3}, Y_{i3}, Z_{i3}) >, i = 1...n\} \tag{3}$$

Eqn. (2) and Eqn. (4) show the mathematic models of a point cloud P_C and a tessellated CAD model of a gauge. G_i represents a triangle with three vertexes (X_{i1}, Y_{i1}, Z_{i1}), (X_{i2}, Y_{i2}, Z_{i2}),and (X_{i3}, Y_{i3}, Z_{i3}). Symbols m and n are used to represents the total number of points in the point cloud and the total number of triangles in the CAD model. Usually, m is much larger than n, that is because m is related to the resolution of the area sensor, and n is the number of triangles tessellated from the CAD model of the calibration gauge. An ICP-based method is developed to determine the transformation matrix T_{Ci}^{Pi} according to these two data set P_C and S_M.

The developed ICP-based method is an iterative process, and each loop has two steps:

1) For a point in the set Pc, find a closest point n_k in the CAD model. When using the triangle representation, a point set P_M can be extracted from the CAD model, $P_M = \{n_k \in G_i, G_i \in S_M, i = 1...n\}$, where n_k represents a 3D point.
2) Compute a transformation matrix T_i such that for a series of points p_j, the moved points $(T_i \cdot P_j)$ are close to their correspondent points n_j in set P_M, with a cost function:

$$\Delta_i = \sum_{j=1}^{M} \|T_i \cdot (p_j) - n_j\|^2 \tag{4}$$

Eqn. (4) can be minimized in the least squared sense. Once p_j and n_j are determined, a quaternion based method can be applied to calculate the matrix T_i. Subsequentially, a new series of points n_j can be found. This process is iteratively executed until the function Δ_i converges to a predefined threshold.

An ICP-based method usually works on two sets of data that have similar format. For example, both are sets of 2D or 3D points. However, to register a point cloud to its CAD model, we have one set of 3D points and one set of triangles. Registration will not work until we convert the triangle set to a point set. Therefore, a sampling method is introduced as following.

As shown in Figure 2(a), T_1 and T_2 are two triangles in a tessellated CAD model. Point $p1$, $p2$ and $p3$ belong to a point cloud. Points $n1$, $n2$ and $n3$ are extracted from the CAD model that are correspondent to $p1, p2$ and $p3$. If a projection point falls into a triangle, then the closest distance is calculated as a point-to-plane distance, such as $p2$ and $p3$. However, for point $p1$, if the projection point falls out of a triangle, then point $np1$ is determined such that a point-to-point distance $p1-> np1$ is used as the closest point distance.

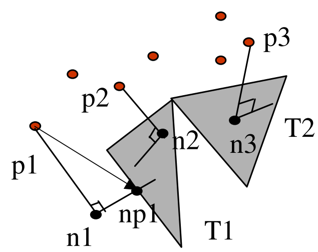


Fig. 2. Closest distance of a point to a triangle

This ICP-based algorithm can be summarized in Figure 3. Usually, since the threshold of Δ_i is not easy to predefine, the system converges when the change of Δ_i, δ, is close to 1:

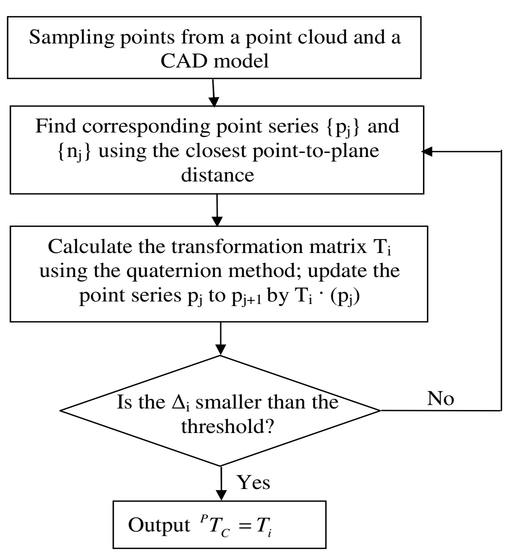


Fig. 3. The flow chart of an area-sensor-based robot hand-eye calibration using an ICP algorithm

$$\delta = \frac{\Delta_i - \Delta_{i-1}}{\Delta_{i-1} - \Delta_{i-2}} \times 100\%. \tag{5}$$

Once the matrix T_{Ci}^{Pi} is determined, matrix T_{Ci}^{W} can then be calculated, assume the motion platform's repeatability can be ignored, matrix T_{Pi}^{W} will always be fixed as long as the robot/CMM inspection station is not moved. Therefore, any measurements on viewpoint Vi can be registered to world frame by directly multiple matrix T_{Ci}^{W}. In another word, an automatic online point cloud registration is realized.

III. A LINK CLUSTERING ALGORITHM FOR POINT CLOUD FILTERING

Outliers is usually generated by intensity error. When the area sensor loses its focus for camera/projector lens, it can also cause outliers in a point cloud. There are often two types of outliers: one type of outliers exist randomly because of intensity noise or dark spots on part surface. The other type of outliers often stay at the boundary of a shadow region, which usually caused because the limited boundary illumination condition.

The first type of outliers are similar as a pulse function in signal processing. It can be easily removed by many filters using 2D image processing technique [5]. The second type of outliers, which often form a stripe of points, is not easily to be identified and removed.

Since a point cloud is developed from a 2.5D height map, in which the unit of X/Y axis is usually in $pixel$ and the unit of Z axis is in mm. Therefore, filtering techniques of 2D image processing may also be applied for 3D point cloud filtering, as long as the it removes noise without changing other data points. It has to be noticed that many smoothing algorithms cannot be applied because the data points are modified such that the small defects like dints/dents cannot be distinguished any more.

In this paper, we use a link clustering algorithm for identifying and removing outliers. As shown in Figure 4, the clustering algorithm iteratively labels data point in the 2.5D height map till each point is assigned to a class. After the labelling process, all points are classified into groups to form several patches that are "linked" to each other. The link distance threshold between two neighbor points can be adjusted according to the resolution of area sensors. This distance is usually set to $0.1 - 0.5mm$, according to the continuity of surfaces. For a smooth surface, a class of data patch can contains thousands of points. On the contrary, multiple small data patches will be generated on a terrain surface with many steps and corners.

The calculation cost of this link clustering algorithm depends on the surface property, it is usually fast for a point cloud measured on a smooth surface. But even on a terrain surface, the entire process can be done with in 1 minute for a height map with 1024×768 points.

IV. SURFACE-ORIENTED ERROR MAP GENERATION

Error map is a color-coded map that is often developed from its point cloud and CAD model. This color-coded error

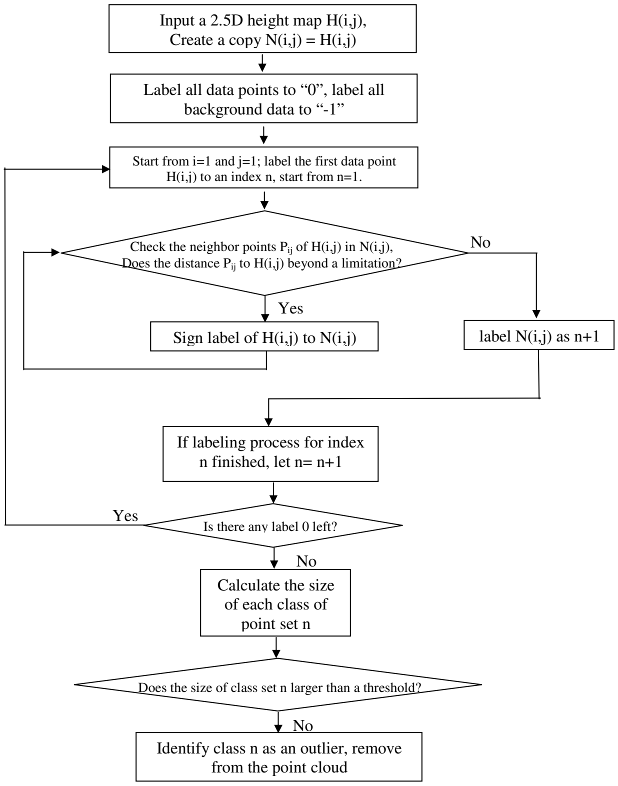


Fig. 4. Flow chart of a link clustering algorithm for outlier filtering

map is actually a modification of Z coordinates: the X and Y coordinates of an error map are same as the point cloud, whereas the Z coordinates of an error map is the distance calculated from a point to its closest triangle. It is necessary to point out that, this color-coded map is used to visualize the dimension differences between the manufacturing surfaces and the design surfaces. It is more emphasize on the relative surface manufacturing error than the absolute shape error. In another word, the color distribution all over the map is more important because it reveals problems in the manufacturing process. Correction of increase/decrease force, temperature, time, voltage/current will be designed accordingly based on this color distribution, or generally called, an error map.

In general, the error of a point is calculated as its distance to its closest triangle, which is determined according to two conditions:

1) The shortest distance of a measured point to the center of a triangle.
2) The projection of the measured point has to be in the triangle.

As shown in Figure 5(a): N is the norm and point E is the center of a triangle $\triangle ABC$, for a measured point $P1$, θ is the angle between two norms and the value equals to $\arccos \frac{N \cdot V_{EP1}}{\|N\| \cdot \|V_{EP1}\|}$. Point D is a projection point of $P1$ on the plane of triangle $\triangle ABC$. This point can be calculated by moving point $P1$ along the reverse direction of triangle norm N by distance $L_{EP1'}$, which is a point-to-plane distance

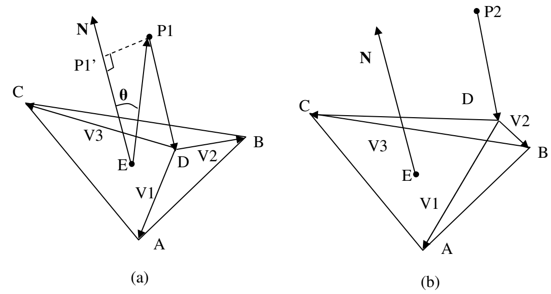


Fig. 5. Closest distance of a point to a triangle. (a) Projection of a measured point falls inside of a triangle (b) Projection of a measured point falls outside of a triangle

calculated by Eqn. (6):

$$L_{EP1'} = L_{EP1} \cdot \frac{N \cdot V_{EP1}}{\| N \| \cdot \| V_{EP1} \|}. \tag{6}$$

Therefore, if point D is in the triangle $\triangle ABC$, the sum of angles among three vectors $V1$, $V2$, $V3$ equals to 2π. If not, point D will be outside of this triangle, as illustrated in Figure 5(b).

Figure 6 illustrates the strategy to calculate surface error. In Figure 6, there are three types of representations of a part surface, (1)a designed surface from CAD/CAM modelling software, (2)a tessellated CAD model that contains triangles for representing the free-form surface, (3)a cloud of points used to represent the measured surface on the manufactured part. The designed surface, though it should be used by definition to derive an error map according to a measured point cloud, is often replaced by a triangulation model in calculating an error map. The reason is that the calculation of the distance from a point to a free-form surface cost more time, considering a large auto part that has millions of points measured for dimension inspection. The approximate error between the free-form surface model and the triangulation model is usually controlled by the dimension of a triangle, the maximum approximation error, the largest distance of a point from a free-form surface to its triangle, can be controlled under as small as to several micrometers. Many triangulation software had implemented this functionality. Therefore, in this paper, the CAD model tessellation error is not included in the error map. The error map is derived according to the measured point cloud and the triangulation model of a part surface.

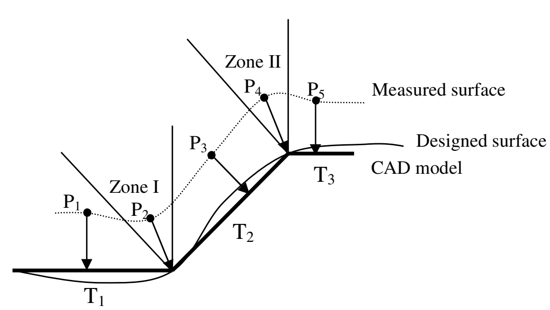


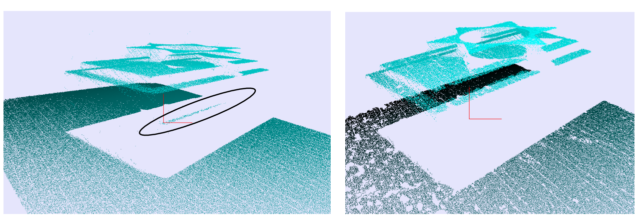
Fig. 6. Calculate error distance of a point to its correspondent triangle

The process to calculate an error map is:

1) Calculate the norm of each triangle of a CAD model.
2) Find the closest triangle of a measured point in the CAD model.
3) Calculate the distance of the measured point to the its triangle.
4) Check the neighbor triangles of the closest triangle: if a measured point falls into multiple triangles, the surface error will then be calculated from the shared triangle boundary to the measured point. For example, point $P2$ and $P4$ of zone I and zone II in Figure 6.

V. Experimental Implementation

The developed link clustering algorithm is tested first on a part with many discontinue surfaces, outliers often exist at the boundary of steps. As shown in Figure 7(a), the circle regions contains groups of points that need to filtered out. Figure 7(b) illustrates the filtering result. It can also be seen in Figure 7(b) that some data points may also be removed because the small size class criteria. For point cloud registration, it is allowed because a point cloud usually contains thousands of points, the outliers may bring the transformation error, but reducing the number of good points in a certain range will not change the calibration accuracy. The filtered point cloud, however, may not be used for error map generation yet because of those holes.



(a) (b)

Fig. 7. Point cloud filtering using a link clustering method (a) point cloud with outliers (b) filtered point cloud

A robot-integrated dimension inspection system has been developed [2]. Figure 8 shows the implementation of our area sensor prototype on such a system. This robotic system can automatically move the area sensor around the part and output an error map for the quality inspection of the 3D shape.

An inspection is tested on an auto part, pillar(part number, m321510), the testing task is to check the shape error on the top ridge of this pillar. Figure 9 shows five point clouds that are registered together into a world frame.

Figure 10 displays the tessellated CAD model of this part. There are totally 2,716 triangles used to represent the geometry shape. This pillar is made by punch a sheet metal, the force directions are displayed, at the left side, the given force is not evenly distributed, which may cause error by pushing the surface up.

Fig. 8. A robot-aided 3D dimensional inspection system

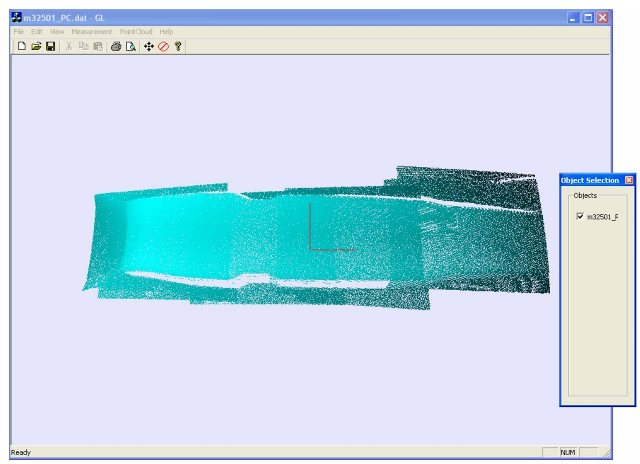


Fig. 9. Registration of point clouds in the world frame of the inspection system

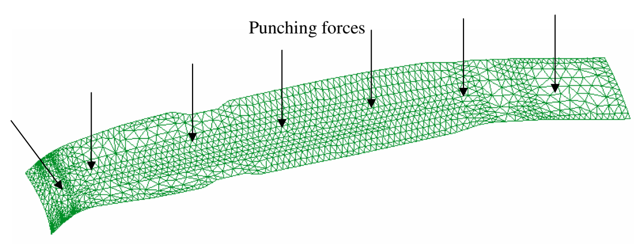


Fig. 10. CAD model of an auto part, pillar(m32510)

Figure 11 illustrate an error map, the red color represents that the part surface is higher than the designed shape, the blue color represents that the part surface is lower than the designed shape. This error is generated by the concentration of the punching force to the part surface. And this tells the designer, even the die has a correct shape, a fabricated part may not have desired shape because the other factors in the manufacturing process. According to this error map, either the manufacturing process or the die need to be corrected such that the shape of this part meets the quality specifications.

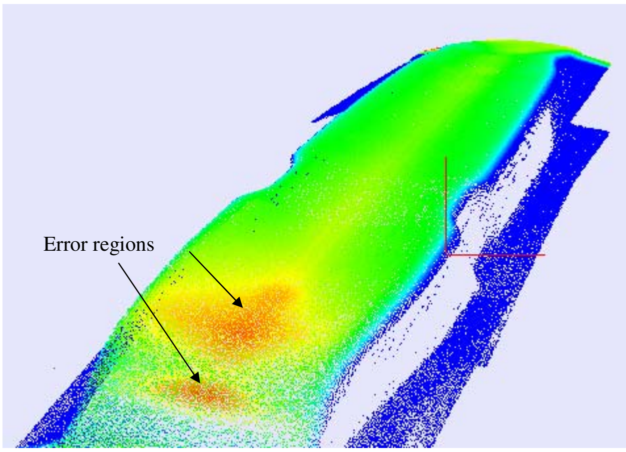


Fig. 11. A color-coded error map of the measured pillar surface

VI. Conclusion

This paper introduces our automatic data processing methods for a robot-aided 3D shape dimension inspection system. Because an automated system is preferred in industrial, the measured point clouds on each viewpoints need to be automatically stitched together in a common coordinate system. Without using robot kinematics information, a new registration method, is developed in this paper that transfers point clouds to a world frame instead of the robot base frame. This method has a better registration performance because robot repeatability is often better than robot accuracy. Therefore, as long as the robot can move the area sensor to a target point in tolerance, the transformation based registration method can quickly stitch all point clouds together. In this paper, a link clustering method is also presented to identifies and removes outliers with keeping small surface defects such as dints and dents for inspection. The number of labelled points in a clustered patch, is used to determine if this patch is outliers or not. Once the point clouds are cleaned and registered to its CAD model, methods for calculating an error map are described as well. By ignoring surface tessellation error, a point-to-plane distance is used to calculate distance between a point to its closest triangle, which forms a surface shape error map. System setup and experimental results are presented in this paper.

VII. ACKNOWLEDGEMENT

The work described in this paper is partially supported under NSF Grant IIS-9796300, IIS-9796287, EIA-9911077, DMI-0115355, and FORD Motor Company University Research Program. The authors also would like to acknowledge the technical assistance and support to this project from Dr. Yifan Chen of Ford Motor Company.

References

[1] Q. Shi, N. Xi, H. Chen, Y. Chen, *Integrated Process for Measurement of Free-Form Automotive Part Surface Using a Digital Area Sensor*, IEEE International Conference on Robotics and Automation, pp. 580-585, 2005.

[2] Q. Shi, N. Xi, H. Chen, Y. Chen, *Calibration of Robotic Area Sensing System for Dimensional Measurement of Automotive Part Surfaces*, IEEE International Conference on Intelligent Robots and Systems, pp. 3898-3903, 2005

[3] Q. Shi, N. Xi, Y. Chen, W. Sheng, *Registration of Point Clouds for 3D Shape Inspection* IEEE International Conference on Intelligent Robots and Systems, pp. 235-240, 2006.

[4] P. Besl, and N. Mckay, *A Method for Registration of 3-D Shapes,* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, No. 2, 1992, pp. 239-256.

[5] L. G. Shapiro, and G. C. Stockman, *Computer Vision*, Prentice Hall, 1^{st} edition, January 23, New Jersey, 2001.

[6] H. Zhuang, and Z. S. Roth, *Camera-Aided Robot Calibration*, CRC Press Inc., 1996.

[7] X.C. Zhang, J.T. Xi, and J.Q. Yan, *A methodology for smoothing of point cloud data based on anisotropic heat conduction theory* The International Journal of Advanced Manufacturing Technology, Vol. 30, pp70-75, 2006.

[8] S. Ferrari, G. Ferrigno, V. Piuri, and N. A. Borghese, *Reducing and Filtering Point Clouds With Enhanced Vector Quantization* IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 18, NO. 1, JANUARY 2007.