A Simulated Surface Quality Inspection System for Stamped Panels

Bo Jiang¹, Wuhua Yang², Xinguo Liu¹

1. State Key Lab of CAD& CG, Zhejiang University
2. Manufacturing System Research Lab, General Motors Global R&D

Abstract: Surface quality of body exterior is an important contributor to perceptual quality of a vehicle. To ensure a high fidelity finish of a stamped panel, the local panel distortion or imperfections of the surface due to the forming stresses are usually manually audited by the inspection experts under the structured lightings. In order to identify and minimize the surface quality issue in an early stage of product development, it is desired to integrate a virtual inspection system with the forming analysis. In this work, a simulation framework is developed to mimic the Green Room inspection environment. To simulate the light reflection, a ray-tracing based method is introduced. To improve the alignment between the panels for analyzing the reflection patterns, two registration methods are developed. The validation studies showed that the proposed system is able to identify and visualize the local surface distortion area in stamped panels produced by the forming simulation analysis.

Key words: Surface quality inspection, stamping distortion, ray-tracing, virtual manufacturing

Introduction

To evaluate the perceptual quality of a vehicle, surface quality of the body exterior is always an important contributor. In the early days of the vehicle industry, most vehicle exterior panels are regular shapes, which are generally easy to manufacture. However, as the fair-shaped design becomes popular in the recent consumer market, the production technologies for these panels become more and more complex. In practice, to ensure a high fidelity finish of the stamped panels, surface distortion or imperfections due to forming stresses are commonly audited by manual expert inspection under structured lighting.

Manual inspection is, however, time consuming and expensive. To identify and minimize surface quality issue during the early stages of product development, it is desirable to develop virtual surface quality inspection system and integrate it directly with forming analysis. With the rapid development of CAD/CAE technologies, many researchers have applied the various computer graphics methods into the inspection procedure. One of straightforward method to mimic the structure lighting effects is to print a specific pattern onto the virtual panel and visualize it on screen.

This paper introduced a simulation framework to mimic Green Room inspection environment. This framework is flexible to handle multi panels for inspection and user interactions. It is also designed to have the ability to incorporate multiple inspection algorithms. In this paper we focus on the structured lighting algorithm. Different from the methods that assume a set of pre-aligned panels [21], we use Shape Matching algorithm [4, 21] to automatically register the panels. In order to achieve both real-time interactive performances and accurate analysis of the structured light reflection, we make use of the state-of-the-art GPU (Graphics Processing Unit) based techniques to synthesize the reflection images. The reflection images are calculated entirely on GPU with thousands of threads in parallel. Ray-tracing based on CPU [8, 22] is much slower than the method we proposed, especially for large size panel (indicate the number of nodes of this panel, the same below) and multiple-panel inspection simultaneously. GPU based Environment Mapping [1] is also very efficient, but it is not accurate enough.

The main contributions of this paper include:
- A shape matched based method for the alignment of the candidate panels with spring-back deformation.
- A GPU based synthesis algorithm for the reflection images.
- A flexible simulation framework to mimic Green Room environment.

The rest of this paper is organized as follows. In Section II, we briefly review the previous work related panel inspection methods. Then we give an overview on the processing pipeline of our system in Section III and the pre-alignment and reflection calculation algorithms in Section IV and V respectively. Then we show some validation studies on both artificial and forming simulation panel in Section VI. Finally, we conclude with some discussion in Section VII.

1 Related Work

The panel inspection problem has long been investigated in industry. In this section, we will briefly review some previous work related to our research.

1.1 Traditional inspection methods

There are two main types of inspection methods: contact and non-contact [27]. For a vehicle body panel inspection, non-contact equipment is commonly used. In recent years, as the rapid development of the optical measurement techniques [6], optical based inspection methods have been developed [19], which are in the non-contact category. For example, Liu et al. [14] measure the reflected point of the light along a line on a mirror-like surface directly. Andersson [1, 2] used optical equipment to test defect prediction algorithms. The traditional reflection lighting combined with advanced optical techniques is also very popular. Some commercial companies develop analogous systems [23].
1.2 Simulated inspection methods

There are mainly two categories of simulated inspection methods: direct inspection and indirect inspection.

Direct inspection examines the geometry information of obtained panels directly. Inspectors can check the spatial coordinate deviations of the panel with respect to the designed one. Traditional mathematical concepts such as normal (first-order derivative) or curvature (second-order derivative) can be also used to judge the deviations between the panels. Curvature based inspection methods for free-form surface can be found in [11,13,14]. Kase et al. [16] uses both the change of principal curvatures and the aggregate normal vectors to conduct inspection on free-form surface. Discrete meshes obtained by 3D scanner or FEA can also benefit from these mathematical properties. Meyer el al. [20] proposed a set of discrete differential-geometry operators for triangulated 2-manifold surfaces. The curvature values are usually visualized using color maps. [7] However, discrete curvature calculation is not very reliable due to its dependence on the tessellation of the mesh, and is sensitive to mesh noise.

Indirect inspection uses some pattern drawn on panels to guide the inspection. Depending on the pattern generation method, we have isolines, reflection lines, highlight lines, etc. The references [11,14] give a comprehensive survey on these methods. For discrete form of panels, we refer the readers to [10,15].

Besides the aforementioned approaches, there area few interesting works related to simulated area. Eichhorn et al. [9] proposed a learning approach to take advantage of the experiences of the experts. They ask the experts to mark the defect area, scan them and use soft-computing techniques to classify these different kinds of surface deviations and then identify the new ones. However, building this dataset still needs manual interaction, thus restrict the dataset itself rather small and further affect the learning procedure. Tosun et al. [24] proposes to use reflection lines for mesh optimization and editing. This method can be used to investigate a quantitative metric for reflection lines based inspection.

The work closely related to our study is presented in [22]. They use cylinder light source and ray-tracing to simulate the reflection. However, they did not mention the method to align panels and finding the corresponding points is time consuming.

1.3 GPU based rendering

Programmable GPU allows users to write their own programs (shader program) to perform various computation tasks. In this work, we utilize the shader program to calculate reflection in parallel. For a more comprehensive understanding of modern graphics and GPU techniques, reader can refer to [5].

2 System Overview

The main purpose of our system is to simulate the whole structured light inspection procedure in desktop PC. As illustrated in Fig. 1, we fix the position of light source and panel, and change the position and the direction to observe the reflection pattern on the panel, which is similar to the real case in practice. The relative motion between the camera and the panel will not affect the result, thus is not taken into account from now on.

![Structured Light Pattern](image)

**Fig. 1** Schematic diagram of computational domain

The pipeline of our inspection framework is shown in Fig. 2. First, the users load the panels (four in the figure) into the system. Then we do registration for inspection task that contains more than one panel. After the panel alignment, we calculate reflection pattern on each panel using shader programs which running entirely on GPU. Then users can interactively change the pattern of light to reveal the potential local surface distortion.

![Model Input](image)

![Registration](image)

![Inspection](image)

**Fig. 2** The panels (two or more) are registered with reference to a selected one. A reflection pattern is then rendered a region of interest for inspection

The GUI (Graphics User Interface) design of our system follows common commercial software and is shown in Fig. 3. Due to the limitation of screen area and requirement of inspection precision, we normally load up to four mesh panels and inspect simultaneously. However, number of panels to be inspection can be dynamically adjusted as needed.
3 Registration Method

When inspecting only one panel, as the virtual camera and virtual light source are both fixed in a single shot, once we fix the position of the panel, we can get reflection pattern on the panel and do further analysis. However, when inspecting two or more panels simultaneously, it is a little bit complicated. We need a registration procedure for two reasons. First, scanned results from different scanning shots are not well aligned, so we need to bring them nearby. Second, since reflection pattern is sensitive to panel position, in order to compare the light pattern in the same region between different panels, we have to further refine the registration result and do registration based on local regions. In our implementation, we always fix one panel, and register the rest according to the fixed one.

In order to align the panels, we use Shape Matching method\(^{[4,24]}\). Since the panels tested this study have same node correspondence, we do not need the Iterative Closest Point (ICP) algorithm\(^{[5]}\). For panels that have different mesh topology, the sampling procedure and ICP can be applied.

Let \( \{ p_1, p_2, \cdots, p_n \} \) and \( \{ q_1, q_2, \cdots, q_m \} \) be the two node sets of candidate panels, where \( p_i \) corresponds to \( q_i \). To find the global rigid registration transformation of the two point sets, we minimize the following energy function.

\[
J(R, t) = \sum_{i=1}^{n} \| R p_i + t - q_i \| \tag{1}
\]

where \( R \) is a 3D rotation matrix and \( t \) is a 3D translation vector.

To solve above optimization problem, we use Singular Value Decomposition (SVD) to find \( R^{[4,21]} \), and then \( t \) is handy. Once we obtain the rotation matrix \( R \) and the translation vector \( t \), we can transform one panel towards the other one. For a set of panels, commonly with one design panel and several tryout panels, we transform all panels towards the designed one.

Since the aforementioned registration is rigid, and complex deformation due to spring-back exists in panels, we cannot get a perfect registration with any global rigid registration. Non-rigid registration is a solution but it is not an easy task to incorporate the complex deformation procedure into simple rules. There is multi-scale nature exists during inspection, i.e., large scale for coarse inspection to locate defect region and small scale for careful examination, we choose to register locally at ROI (Region of interest).

The formulations of this local rigid registration are exactly the same as aforementioned global rigid registration. We can select only coordinates of nodes in ROI specified by users into the registration procedure, see Fig. 4.

4 Reflection Calculation

Given the registered panels, we now “print” reflection pattern on them. We will first derive the formulations of the reflection, and then implement it on GPU.

4.1 Mathematical Formulation

In the context of structured lighting, all the reflections we concern about are the first bounce reflection of the structured light source. Multi-bounce reflection and the ambient light are both ignored in this study.

As shown in Fig. 1, when it is viewed from position \( E \), the unit vector for viewing direction is \( V = \frac{E - P}{\| E - P \|} \). Then the reflection direction can be calculated by reflection law:

\[
R = 2(V, N)V - N \tag{2}
\]

where \( N \) is the unit vector of the surface normal at point \( P \) and \( \langle V, N \rangle \) is the dot product of vector \( V \) and \( N \).

Now we need to compute the intersection point of the reflected ray \( R \) with the light plane. Let \( \{ L, L_x, L_y \} \) be the local frame of the light plane, and \( W_x, W_y \) be the size (width and height) of the light plane. Let \( Q \) be the intersection point of the reflection ray with the light plane. Then we can represent intersect point \( Q \) as:

\[
Q = P + tR \tag{3}
\]

where \( t = \frac{\langle L - P, L_x \rangle}{\langle R, L_x \rangle} \) is the parameter of \( Q \) on the reflection ray.

In order to get the color of intersection point, we need to know the coordinates of \( Q \) in light plane local frame. In fact, we can also represent \( Q \) as:

\[
Q = L_x x + L_y y \tag{4}
\]

where \( L \) is the local frame origin, \( L_x \) and \( L_y \) are the two coordinate axis perpendicular to each other and \( (x, y) \) is the local coordinate of \( Q \) in the light plane.
Then we can get the coordinate values as:

\[
x = ( ( tR - (L - P) ), L_x )
\]

\[
y = ( ( tR - (L - P) ), L_y )
\]

(5)

(6)

After compute the local coordinate \((x, y)\) in the light plane, we can convert them to the texture coordinate by dividing them by the light plane size

\[
t_x = \frac{x}{W_x}
\]

\[
t_y = \frac{y}{W_y}
\]

(7)

(8)

Now, we can use the resulting texture coordinate \((t_x, t_y)\) to look up color in structured light texture.

4.2 Reflection Calculation on GPU

As we can see in the previous section, the reflection calculation for all panel nodes is totally independent on each other, which can be easily parallelized. Recalling in Section II that GPU is specialized in highly parallel tasks, thus we can utilize the computation power of GPU to perform the reflection calculation, especially for very large panels or multi panels inspected simultaneously. Fig. 5 shows the pseudo-code of the shader program

```cpp
bool intersect(vec4 rayStartPoint, vec3 rayUnitOrientation, vec4 planeCoefficient, inout vec3 intersectPoint)
{
    Calculate reflection ray;
    Calculate intersection point between reflection ray and light plane;
    if (intersection point exists)
    {
        intersectPoint = point;
        return true;
    }
    else
    return false;
}

void main()
{
    intersect(rayStartPoint, rayUnitOrientation, planeCoefficient, intersectPoint);
    Use intersect point on light source to look up color;
    Update color in this pixel;
}
```

Fig. 5 The pseudo-code of the shader program

5 Implementation and Results

In this section, we will present some analysis results based on both artificial and industrial stamping panels.

All simulations are tested on an HP® mobile workstation with Intel® Core™ 2 Duo CPU T9500 @ 2.67 GHz, 4 GB RAM, Nvidia® Quadro® FX 3600M mobile graphics card, and Microsoft® Windows® 7 Enterprise (64-bit). The system is tested with both 32-bit and 64-bit compilation

5.1 Artificial Panel

In order to verify our algorithm, we test it on a set of artifi-

cial panels, shown in Fig. 6. Top left is the original panel (a segment of cylindrical surface). The remaining three (from top to bottom, left to right) are panels with artificial defects (sinusoidal wave, gradually increased amplitude) in the middle of these panels.

We can obviously see from the results that the light pattern faithfully delivers the information we need, i.e., where the defects are and the level of these defects.

![Fig. 6 Reflection pattern of a cylindrical surface with artificial defects](image)

5.2 Industrial stamping panel

Besides artificial panels, we also test our algorithms on a set of vehicle body panels shown in Fig. 7. Two panels are drawing mesh and spring-back mesh from stamping FEA respectively. Since the rectangle pattern frame (in yellow) can be resized and moved

![Fig. 7 Inspection results of stamping simulation panels](image)

arbitrarily, we can select a targeted area for inspection and comparison study as indicated by obvious differences in reflection patterns between two panels.

We also notice that the density and direction of the stripe pattern of the light source will significantly affect the final reflection pattern on panels. Fig. 8 shows two reflection patterns on punch surface (left) and the spring-back panel (right). The local reflection pattern is less distorted under horizontal stripe as compared to vertical stripe. It is an interesting task to automatically
find particular view angle and specific light pattern (direction, density, etc.) that can maximize the image difference in reflection pattern.

6 Conclusion and Discussion

In this study, we developed a simulated surface quality inspection system to mimic Green Room environment. It is a complete virtual environment for panel inspection under structured light. This analysis system is able to identify and visualize potential local surface distortion in stamped panels based on forming simulation results, thus can greatly reduce the engineering cost associated with physical panel tryout.

By global and local rigid pre-registration procedure, the system offer the flexibility to work on the region of interesting and improve local registration accuracy. Simple GUI and high utilization of GPU make the system easy to use and achieve real-time performance for multi panel inspection.

In future, we plan to incorporate a robust non-rigid registration method into the inspection. Non-rigid registration can further eliminate the issue related to panel spring-back and achieve a better alignment result. Since the panel inspection task is subjective and has low repeatability, it is necessary to develop a quantitative method to better understand the relationship between human visual perception and underlying reflection pattern for a fully automatic inspection.

Acknowledgement

The authors would like to thank Robert Tilove of General Motors R&D, Meng Luan and Ruyang Shou of Zhejiang University for many helpful discussions during this study.

References


