MVA2015 IAPR International Conference on Machine Vision Applications, May 18-22, 2015, Tokyo, JAPAN

# Development of Design and Operation Supporting Techniques for Product Inspection Devices Using Virtual Devices

Fumiyuki Takahashi [1], Hideki Abe [1], Tetsuo Koeduka [1], Susumu Haga [2]
[1] FUJITSU LABORATORIES Japan, [2] FUJITSU Japan
{Takahashi.fumiy, abe.hideki, koezuka, shaga} @jp.fujitsu.com

## Abstract

*We have developed the virtual system for product appearance inspection device. The system can replicate the state of the real device including the illumination and the camera. The essential parts of this system are the optical simulation using reflection mapping technique and calibration of virtual system with real device. We will show example of the system being applied to real device operation.*

## 1. Introduction

With advancing digitization in the manufacturing field in recent years, there is a movement to try to improve the product lifecycle management (PLM) in efficiency by utilizing product design data not only for design and engineering processes but also for prototyping, manufacture and other processes. Despite this movement, no papers have been presented to date with respect to the application of design data to product appearance inspection processes for improved operational efficiencies. It is difficult to appropriately make the virtual product appearance inspection device including its special optical systems. For this reason, instruction work of an appearance inspection device is still being conducted by the device operator's manual operation based on intuition and experience. It is considered essential to develop techniques that would improve inspection efficiency and stability by utilizing design information further in the future. This paper introduces our newly developed technique that can virtualize a product appearance inspection device, which examines the condition of product painted surface, including its illumination and imaging systems.

## 2. System in General

Figure 1 shows the product appearance inspection device. This device detects surface defects on the product such as scratches with a high degree of accuracy by projecting multiple light-and-dark patterns with varying phase onto the inspection object's surface, capturing the image by a camera, and processing a multiple number of such images [1]. Detecting regular reflection light from the inspection object, this device can inspect only the surface that faces the illumination and the camera. For this reason, if the inspection object has a complex shape, the device must capture the images of the object in many different orientations, requiring increased time of the instruction work. Figure 2 shows examples of the cap tured images necessary to inspect a mobile phone.

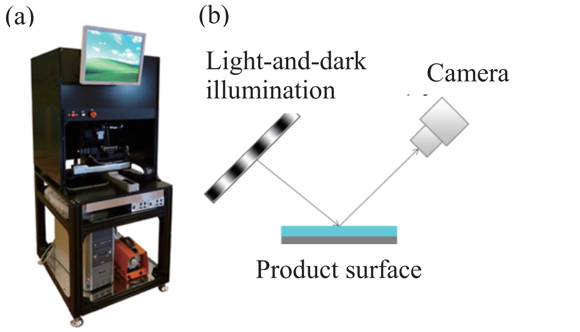


Figure 1. Product appearance inspection device
(a) Appearance (b) Schematic optical system

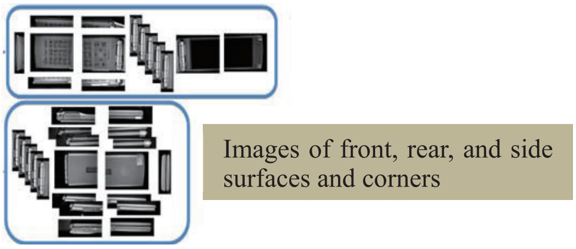


Figure 2. Inspection views (mobile phone)

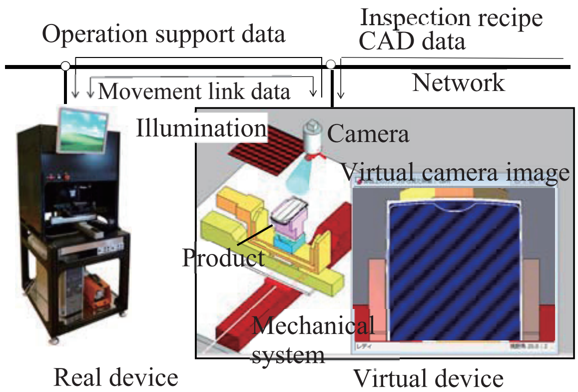


Figure 3. Image of virtual inspection device system

Figure 3 is the image of a virtual appearance inspection device system. The system, which is connected to the network, receives design (CAD) data and inspection recipe, and the virtual device can be generated. In addition to stand-alone operations, the virtual device is also capable of sharing the movement information of the real device and the virtual device on the network, allowing these devices to link their movement to each other. For example, the system replicates the state of the real device in instruction operation on the virtual device in real time, making it possible to transmit the information from the virtual device to the real device to help make instruction operation more efficient. The following sections are descriptions of the essential parts of this system—optical simulation and calibration technologies.

## 3. Optical Simulation

### 3.1 Camera Image Simulation

This section is a description of the optical simulation that replicates the camera image from a product appearance inspection device on the virtual device. Typically, ray-tracing method is used to simulate imaging systems including illumination. However, ray-tracing method, which requires a lot of time for rendering, takes as long as several tens of seconds to several hours to render only a piece of image, making it difficult to utilize such simulation results in real time. This led us to apply reflection mapping technique [2] to our optical simulation. Reflection mapping is a technique widely used in the field of CG to artificially simulate the state of the surface of a reflective object mirroring its surrounding environment. Reflection mapping simulates camera image by first rendering an environment view that was rendered from the position of a reflective object as a texture map. Then the texture map was wrapped around the reflective surface, using texture-mapping method, based on the line of the camera sight and the normal information about the reflective surface. Figure 4 shows an example of a computer simulation using this reflection mapping technique.

### 3.2 Extended Reflection Mapping

Reflection mapping usually maps a point per pixel. In this device, however, if there is a local deformation, such as a defect as shown in Figure 5, light beam emitted from a large part of the light-and-dark illumination enters one pixel due to lens effect. This is an important characteristic utilized in an inspection device to detect surface defects. If this characteristic can be replicated in computer simulations as well, it is expected to be able to broaden the range of application by offering not only device operation support but also the virtual verification of device performance using virtual defects in the device design phase. The use of supersampling for reflection mapping solved our problem mentioned above. Supersampling is an anti-aliasing technique used in the field of CG. More specifically, reflection mapping is performed several times while the position of the virtual camera is moved within a range not exceeding the size of one pixel of the camera, and a piece of camera image is rendered based on the average of those results. This approach allowed the above-mentioned characteristic to be replicated on a virtual system, enabling more realistic computer simulations. Figure 6 shows an example of the application of defect detecting, which is similar to that for the real device, to a virtual camera image. The image generated by extended reflection mapping demonstrated that this approach could simulate the same defect image as the image of real device. In this example, a piece of image is generated from a total of 25 data points, with one pixel divided into five sections in both the width and length directions. Rendering time for an image with a size of 814 x 618 pixels is about 1 second (GPU: GeForce GTX 660).

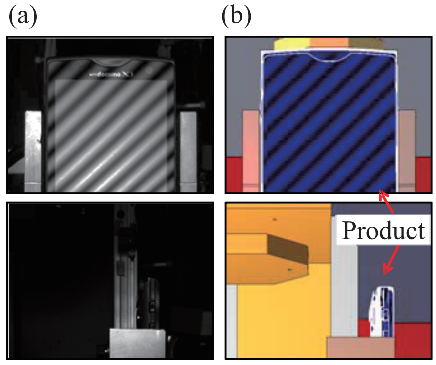


Figure 4. An example of simulation
(a) Camera image of real device (b) Camera image of virtual device

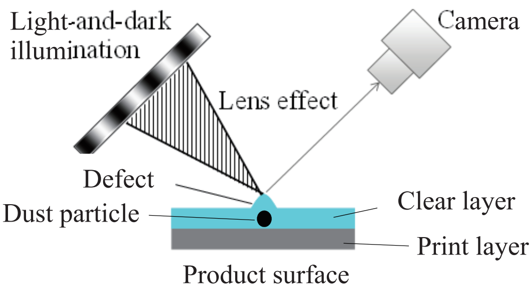


Figure 5. Range of light beam entering single pixel

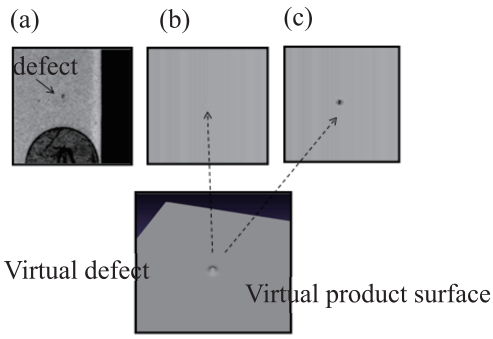


Figure 6. Defect simulation
(a) Camera image of real device (b) virtual device (conventional method) (c) virtual device (developed method)

## 4. Calibration

While the virtual device can be generated exactly the same as design data, the real device cannot be made necessarily to design data. For this reason, after generated from design data, the virtual device needs to be calibrated to the real device without exception. There are three major conceivable error factors in a virtual device that include mechanical and optical systems: mechanical errors (e.g., assembled position and attitude, the original points of actuators, the positional relationship between actuators, feed rate accuracy, the effect of shaft deflection, etc.), optical errors (e.g., focal distance, principal point position, distortion, cell size, etc.), and design data approximation errors (e.g., difference between CAD and real devices). Errors calibrated by the calibration operations described in this section are mechanical and optical errors. Most important with this system is the ability of the virtual device to replicate how the real device's imaging system looks like. In this calibration operation, parameters for the virtual device are calibrated by estimating the optical and mechanical parameters of the real device, with reference to the information about the multiple marker images captured by the real device's camera. The following shows the procedure for calibration.

(1) Capturing image group for calibration

First, the calibration target is mounted on the mechanical system of the real device, and then the camera images that target in several different orientations while the mechanical drive shaft subjected to the calibration is in motion. Note that the position information of all the drive shafts during imaging should be saved as $\mathbf{q}_0$. Figure 7 shows part of the images for calibration.

(2) Optical calibration

First, using a camera calibration technique, the optical parameters of the device camera and marker coordinates $\mathbf{x}_r$ mounted on the calibration target are estimated with reference to the device camera's coordinate system. Then, based on these estimated optical parameters, the virtual camera's optical parameters in the virtual device are calibrated. Note that this camera calibration was conducted using commercial image-processing library HALCON.

(3) Mechanical system calibration

Assuming a mechanical kinematic model, the mechanical system of the virtual device is mounted with the calibration target to determine virtual marker coordinates $\mathbf{x}_v$. Mechanical system parameters are estimated by performing an optimizing calculation to minimize the value obtained from the following objective function of real marker coordinates $\mathbf{x}_r$ and $\mathbf{x}_v$.

$$f(\mathbf{q},\mathbf{r},\mathbf{t}) = \|\mathbf{x}_r - \mathbf{x}_v(\mathbf{q},\mathbf{r},\mathbf{t})\|$$

where $\mathbf{q}$, $\mathbf{r}$, and $\mathbf{t}$, which are the virtual device's mechanical parameters, represent the position of each actuator in the virtual device, the position and orientation of the entire mechanical system, and the position and orientation of the camera, respectively. Among the initial parameters given to an optimization model are the design value and the position of the actuator at the time of imaging $\mathbf{q}_0$. Lastly, the virtual device's mechanical parameters are calibrated using the estimated $\mathbf{q}$, $\mathbf{r}$, and $\mathbf{t}$ to complete calibration. In these mechanical system calibrations, only the errors related to $\mathbf{q}$, $\mathbf{r}$, and $\mathbf{t}$ were calibrated among other mechanical errors, because these errors seem to have larger impacts on the degree of inspection accuracy.

Figure 8 shows the post-calibration camera images of the real device and virtual devices. The figure indicates that the mechanical system's edge positions, the position and orientation of the calibration target, and the optical system's perspectives are consistent. The post-calibration marker position errors between the real and virtual devices did not exceed 0.5 mm, which is accurate enough for use in virtual device applications, which will be described later.

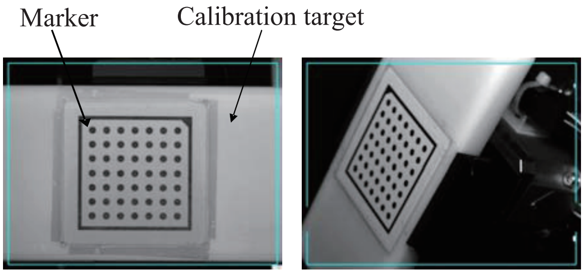


Figure 7. Image for calibration

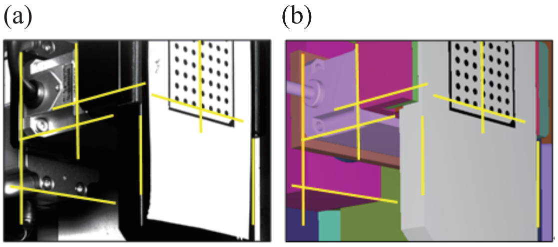


Figure 8. Comparison of camera images
(a) Real (b) virtual

## 5. Examples of Applications

As an application example, here is the case in which this newly developed system improved the efficiency of setting an appropriate illumination conditions for a product appearance inspection device. Setting illumination conditions is an operation to adjust three illumination parameters, that is, illumination intensity and light-and-dark pattern shapes (widths and angles), to an adequate value for each inspection position. The operator moves the inspection object to each inspection position, and fine-tunes illumination parameters to get adequate levels of illumination while checking the captured image. If there are many inspection positions, it takes time for the operator to adjust all illumination parameters in each inspection position. There was also the problem that the accuracy of adjustment varied depending on the operator. There is a need to consider the automation of the task that the operator judges, while checking the image, whether the level of illumination is appropriate or not by having image processing take on this task. High precision automated adjustment for illumination parameters needs to recognize where the inspection object is located in the image. If an automated adjustment was made without recognizing the inspection object, the effect of part of the camera image outside the area actually needed for inspection would disable such an appropriate automated adjustment. As shown in Figure 9, however, the camera image contains several other components that are not inspection objects. For this reason, it is extremely difficult to recognize the inspection object in a camera image only with image information. This problem led us to look at recognizing the inspection object in a virtual image, instead of doing so in an actual image. The recognition of the inspection object in a virtual image is easy if the CAD of the inspection object is available. Figure 10 shows the procedure for adjusting illumination conditions. In the virtual device, the inspection object is recognized using binarized images, with the inspection object and other components differently colored. These binarized images are shared through a network with the real device as inspection object area information, enabling the automated search of appropriate illumination parameters by limiting illumination adjustments to this inspection object area. Figure 11 shows examples of the inspection object being recognized. The inspection object recognized in the virtual device is outlined with a dotted line. According to the assessment of instruction time for 40 pieces of inspection object images, the operator took 56 minutes to decide illumination conditions manually. In contrast, the automation using the virtual device could shorten it by about half to 24 minutes, ensuring the usability of this system.

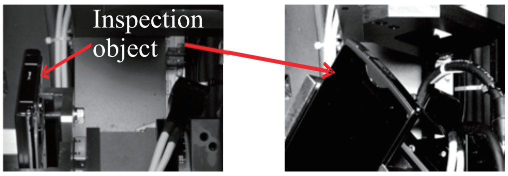


Figure 9. Image of inspection object

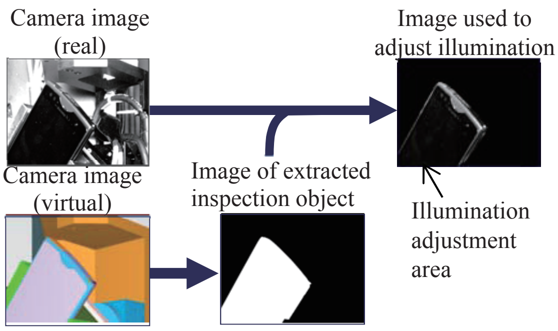


Figure10. Procedure for adjusting illumination conditions

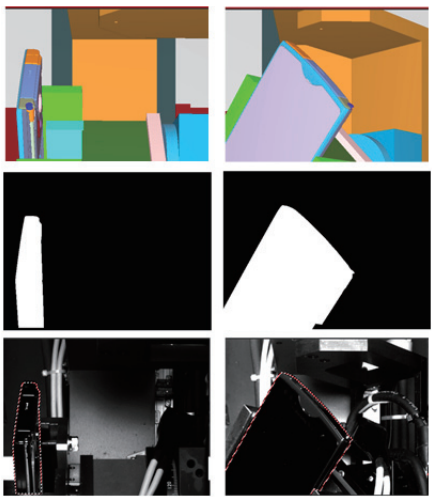

Figure11. Inspection objects being recognized
Camera image in virtual device (top), Image of masked inspection object (middle), Camera image in real device (bottom)

## 6. Summary

The virtualization of a product appearance inspection device has proven that it improves the efficiency of instruction operation for the real device and is an effective tool for the virtual verification of device performance. This study applied the virtual-device-based automation to deciding illumination parameters, but instructing other operations, such as deciding inspection windows, still rely on human operations. It is our intent to continue to study the automation of additional instructing operations. Our future studies would include virtualization techniques for general inspection devices other than appearance inspection devices.

## References

[1] T. Nagato , T. Fuse and T. Koezuka : Defect inspection technology for a gloss-coated surface using patterned illumination, Proc. SPIE 8661, Image Processing: Machine Vision Applications VI( 2013)

[2] J. Bllinn, M. Newell : Texture and reflection in computer generated images, Communications of the ACM, 19, 10,(1976),542