

# GPU-based discrepancy check for 3D fabrication

Fu Che Wu

Providence University  
200, Sec. 7, Taiwan Blvd  
Taiwan, Taichung City 43301

fcwu@pu.edu.tw

Andrew Dellinger

Elon University  
100 Campus Drive  
USA 27244, Elon, North Carolina

adellinger@elon.edu

## ABSTRACT

A GPU-based visualization approach is presented to show the discrepancies between the 3D model and the fabrication of that model. To show the differences, a 3D scanner is constructed to scan the fabrication for comparison with the 3D model. To compare the detailed differences, a high resolution camera with a projector is used. To demonstrate its application, a sculpting assistance system is implemented. The system can capture the three-dimensional model iteratively and provide information by rendering different colors on the surface to show the topological and geometric differences between the final target model and the current model. Then the user can see how to modify the current model to best approximate the final target model. The topological difference is obtained by rendering on the screen the 3D data from both the final target model and the current model. The user can manipulate and observe their differences. The local geometry is compared in the GPU and rendered on the real scene. Users can easily see the geometry directly on the fabrication. To keep a fixed relationship between the 3D model and the fabrication, a reference image is tracked at the bottom stage to recover the related transformation.

## Keywords

spatial augmented reality, 3D scanning, calibration, registration, fabrication

## 1. INTRODUCTION

For augmented reality applications, a discrepancy check is an important feature to accurately match a 3D model to a scene [Kah10a]. For our application, assisting fabrication, a discrepancy check is more important to help the user find the differences between the 3D model and the fabrication. For example, the design process of a vehicle begins with designers sketching ideas on paper. From there, the team designs the vehicle in scaled-down clay models and 3D CAD drawings, eventually milling full-size clay vehicles to analyze body styling options. Our interest is in visualizing the discrepancy between a fabricated item and its 3D model.

Art is a creative process. Can it possibly be supported by a computational tool? Flagg et al. [Fla09a] argue that capture and access technology can provide a key form of computational support for the creative process. They have designed an interactive system for guiding artists to paint and sculpt using traditional materials and tools. In their system, a camera is used to capture the current process and a projector is used to guide the user how to progress. We constructed

this system and evaluated how useful it is in guiding a novice sculptor in reproducing a work in a clay medium.

It is challenging for a beginner to sculpt from scratch. As a guideline, a sculptor should first use a pen to draw a shape on the target. To reproduce a work from a different viewpoint, the ratio scale and the position should stay as correct as possible. However, this is not easy. The purpose of our system is to fill the gap. Through the projector, the system renders the guiding information on the surface to let the user know where to carve. Red portions of the surface show the sculptor where to add clay. Blue portions of the surface show the sculptor where to remove clay. The key technology in implementing such a system is 3D scanning and projector-based rendering. Both technologies rely on an accurate calibration process. 3D scanning technology is increasingly developing, both in terms of accuracy and affordability. A projector and a camera are enough to construct a 3D scanner system. The price is about 2000 U.S. dollars or less. Projector-based rendering also received significant notice recently, since it is not only suitable for large areas of display, but also for rendering on arbitrary surfaces. Thus, the combination of 3D scanning and projector-based rendering is ideal for constructing a Projector-Camera System (Procams) for sculpture assistance.

Rivers et al. [Riv12a] demonstrated the ability of the system to aid in creating an object that matches a 3D model, sculpting from a physical example, and making a stop-motion animation. Their system

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

provides depth guidance and edge guidance. To show the guidance information to a user, the system just calculates the difference between the target model and the current state. Shape matching includes topological and geometric similarity. In addition, 3D manipulation is a challenge for 3D interface designers, since it involves the control of six degrees of freedom. However, rendering the difference on the surface only considers the geometric variation. Thus, during the sculpting process, how to construct the correct topology becomes the hardest part. If the topology is misaligned, comparison mechanisms will show great differences between the 3D model and the sculpture, even though their shapes are similar. To solve this situation, our system not only performs guidance rendering on the surface, but also renders the 3D structure on the screen to help the user recognize the topological relationship between the target and the current state of the sculpture.

The system renders guidance information on an extended desktop view (Figure 1). The left side image shows the 3D models that will be displayed on the screen. The right side image shows the differences between the current sculpture and the target. From our observations, monochromatic displays of difference in clay depth are easier to use than a spectrum of colors dependent on the amount of difference in clay depth.

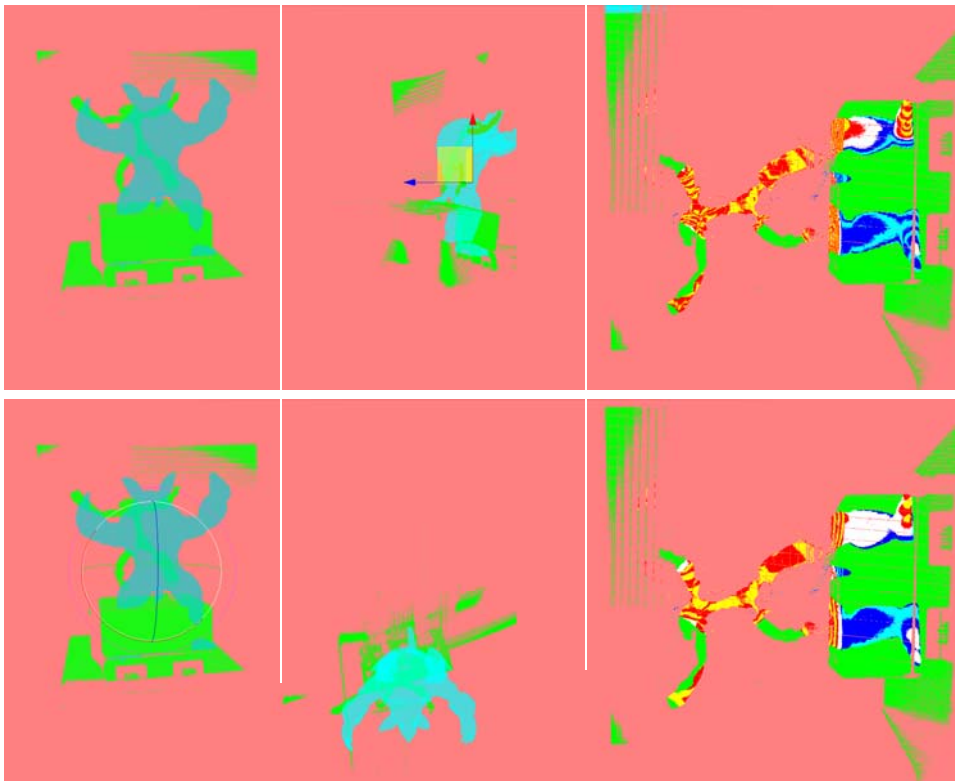
In our system, there are some functions that are

different from previous systems.

- 3D structure is captured and rendered on the screen, and manipulator operators are provided.
- The difference between the target and the sculpture is indicated monochromatically, with a threshold value to adjust its cover range.
- The comparison mechanism is implemented on a GPU.

## 2. PREVIOUS WORK

Using the projector as a guiding tool has already been demonstrated in many applications. The idea is to use projected imagery to illuminate physical objects, dynamically changing their appearance. The idea inspired us to develop such a system. In the past, Raskar et al. [Ras01a] demonstrated changing surface characteristics such as texture and specular reflectance, as well as dynamic lighting conditions, simulating cast shadows that change with the time of day. The concept was extended to dynamic shade lamps [Ras03a], whose projected imagery can be interactively modified, allowing users to "paint" synthetic surface characteristics on physical objects using a tracked wand [Ban01a]. Also, Raskar et al. developed a system to provide extra information about the environment [Ras04a]. The system consists of a projector with a camera. When the camera captures the marker in the environment, the projector can provide the related information. Flagg et al.



**Figure 1: Rendering images with different threshold values**

[Fla06a] used a projector to guide the user in learning how to paint by projecting the painting on the canvas. Therefore, the user can easily determine the position and size of each element. Löchtfeld et al. [Löc11a] used a mobile projector mounted on the headstock of the guitar. By projecting instructions directly onto the strings of the guitar the user is easily able to realize where the fingers have to be placed on the fretboard to play a certain chord or a tone sequence correctly.

In these applications, the key technology is the calibration. To solve this problem, many different approaches are surveyed. A general approach is taken by Sun et al. [Sun08a], who used a chessboard pattern for calibration. To solve the issues with calibration among multiple projectors, Sajadi et al. [Saj09b, Saj10a, Saj10b] assumed there is a vertical projection plane and used the location of the four corners and upper and lower boundaries to estimate the parameters of the projector and camera. First, an approximate location of the camera is inferred from the position of the four corners, then more accurate camera parameters are estimated by the upper and lower boundaries, then finally they can determine the parameters of the projector. After calibrating the location of the projectors, seamless multi-projectors can be formed through color calibration [Saj09a].

Okatani and Deguchi proposed a simple calibration method [Oka09a] that assumes that the internal parameters of the projector, except the focal length, are known. Then, a camera captures the same pattern image that is projected from four projectors. All the parameters can then be calculated. Drareni et al. [Dra09a] have a similar approach. The projection surface is assumed as a plane, then the point on the projection surface with the pixels of the projector can form a homograph relationship. All the parameters can be estimated through the different positions of the projectors. Bhasker et al. [Bha07a] used a low-resolution camera to calibrate a multi-projector projection plane. This method can also calibrate the deformation of the projection surface. Similarly, Kimura et al. [Kim07a] used a calibrated camera and projection plane to calibrate the parameters of a projector.

Griesser et al. [Gri06a] used a handheld whiteboard in front of the camera. By changing the whiteboard's position and angle, different image patterns could be captured. After the corresponding relationship among these patterns is recovered, the parameters of the projector can be calibrated. Our method is similar to their method.

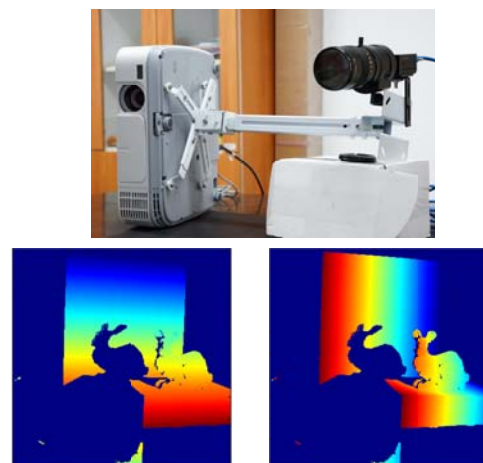
In general, the projection surface is a plane. Thus, the relationship between projection surface and the surface of the projector is a linear relationship. It is easy to calculate; however, Johnson [Joh06a] obtains the optimal solution by using the genetic algorithm to solve the spherical projection surface problem. If the

parameters of the projector came out in the estimation, then its position can no longer be changed. In order to change the position, its parameters need to be re-estimated. To solve this problem, Zhou et al [Zho08a] combine the projector and camera together. Certain feature points in frames will be discovered. Based on these feature points, the change of the location of the projector can be estimated. In this manner, the purpose of continuous calibration can be achieved. Ishihara et al [Ish07a] proposed another method in which no feature point is needed to continuously calibrate the projector. In this method, by measuring the changes in the projection position and intensity, the location changes can be estimated. Dao et al. [Dao07a] used a 3D tracker to continuously track the projector in the space of change, which, in turn, can make a handheld projector maintain a fixed image in the projection plane. By using an angle-changing mirror, Choi et al. [Cho07a] made the system respond to the changes of angle with corresponding calibration.

### 3. PREPARATION ENVIRONMENT

The system used in this study consists of a projector and a camera. The projector is the NEC M300X, which has an effective pixel resolution of 1024x768. The focal length of its lens is 17.5-29mm. The camera is the Point Grey's Grasshopper 3 (GS3-U3-41C6C-C) with a 16-90mm/1:1.7 lens. The resolution of its image is 2048x2048. Preparing a suitable environment is very important. Before setting up the system, we should decide the size of the working area for the statue, and we should adjust the focus of the projector and the cameras to make sure the working area in the image is clear and covers the pixels as much as possible. The working area is situated about two meters in front of the projector. The distance between the projector and the camera is about 30 cm.

For mobility considerations, the camera is tied to the projector. To get an accurate depth value, the camera should be placed as far away from the projector as



**Figure 2: Recovered projector column indices in vertical and horizontal directions under current configuration.**

possible without significantly reducing the overlapping area from the two views. Too much distance makes it difficult to find the corresponding pair, and too little distance will increase the depth error. Figure 2 shows the results of the configurations of the camera and the projector using the corresponding maps in the horizontal and vertical directions. The horizontal map did not provide much depth information due to the strip pattern almost parallel to the connected line of the center position of the camera and projector.

Flycapture SDKs are used to drive the camera. The camera's intrinsic and extrinsic parameters are estimated by the Matlab camera calibration Toolbox [Bou00a]. In a similar way, a projector can be assumed as an inverse camera. To reduce the complexity and improve the robustness of the estimation process, an auxiliary camera is used to recover the 3D position of the projected pattern. Thus, the projector's parameters can be estimated by capturing a structured light pattern from the projector.



**Figure 3: The scanning result of the target model**

The 3D scanner is implemented using Lanman and Taubin [Lan09a] as a reference. After the parameters of the camera and projector are estimated, the 3D depth map can be calculated by projecting the structured light patterns onto the scene.

Thus, we can get the cloud point, the set of target models, and the current state of the working model. Figure 3 shows the scanning result of the target model. We can check that the system is configured properly by shading the model with a projector. The 3D model or shading position will be wrong if the configuration is wrong. That is, if any pixel of the lighting is not in the correct position, it is not possible to light only the model.

To begin a sculpture, we should prepare a scaffolding to support the weight of the clay. After constructing the scaffolding, the 3D structure can be scanned and rendered with a projector, as shown in Figure 4.

When there is a miscalibration, there is no sense of whether the wrong parameters were used, there was a coding error, or something else is incorrect. So, each stage of the configuration of the system should be checked immediately after its completion. We can check more easily using an extra device. For example, when we get the camera's extrinsic and intrinsic

parameters, we are not only projecting back the features onto the original images. We are also measuring the physical scale from the camera to the marker to check that the extrinsic parameter is fine. During the projector calibration, an auxiliary camera was used to check that the projected pattern is in the



**Figure 4: The model is lighted by the projector.**

correct position. In our experience, the more patterns that are captured for each angle and position, the more robust is the estimation of the calibration parameters

To speed scanning, we also considered using the phase-shifted fringe method. Zhang et al. [Zha14a] designed a set of hybrid structured light patterns composed of phase-shifted fringe and pseudo-random speckle. Phase-shifted fringe has high accuracy but needs six frames for one depth map. Pseudo-random speckle has low latency but low resolution. Based on motion detection, their system combines these two sources of information to construct an efficient balance between accuracy and speed during depth sensing. However, this method is highly dependent on the uniform albedo of the projected structured light. But the projector in this study is not a DLP projector and the projected surface is a dark material, so the phase-shifted fringe method was not accurate enough. Instead, the gray code method for 3D scanning was used.

#### 4. DISCREPANCY VISUALIZATION

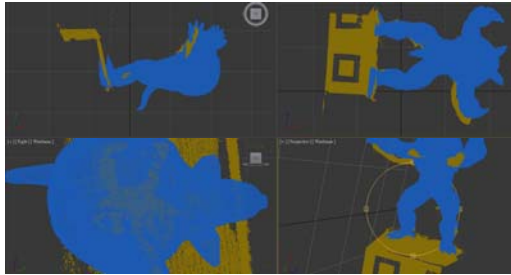
Before comparing discrepancies between the target mesh and the source points, both models should align on the same basis. The initial fitting process can be done in the general modeling tool as shown in Figure 5.

A large amount of computational power is necessary to calculate the difference between the target mesh and the source points because of the huge amount of points involved. Even with an oc-tree structure to reduce the amount of points that should be compared in each run, it is still hard to get results in a few seconds. Thus, a GPU implementation was developed that could show the difference between the two models in real time manipulation.

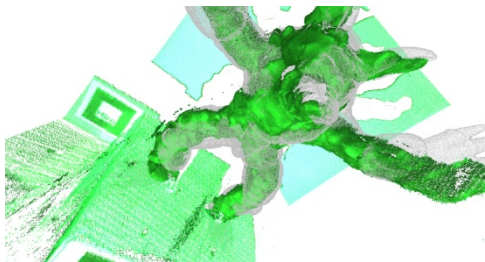


To calculate the difference between the target mesh and the source points, we implement the comparison in the pixel shader program. First, the target mesh is rendered in a depth buffer. Then the program uses the depth buffer as a texture input. Thus, in the pixel shader program, each of the differences between the target mesh and the source points can be calculated at each pixel, as is demonstrated in Code 1.

During the sculpting process, the user should monitor the differences between the source and the target model. However, to determine the correct ratio and scale is not an easy job for a novice user. Each time the user scans the current state of the working model, the system provides two types of information to guide the user in the sculpting process. One type of information is the 3D structure displayed on-screen. The user can change the viewpoint to recognize the difference between the two models, as Figure 6 shows. The second type of information is the calculated difference between the source and target shown by the projector. These differences are projected onto their exactly corresponding positions on the current state of the working model and are shaded with different colors to indicate the type of difference between the two models.



**Figure 5: Initial alignment fitting of two models in the modeling tool**



**Figure 6: The user can recognize the difference between the current state and the target model on-screen.**

To guide a user in sculpting a model, the system will show the differences in different colors. The parts of the model with an insufficient clay depth will be rendered as red, and the parts of the model with too much clay depth will be rendered as blue.

---

```

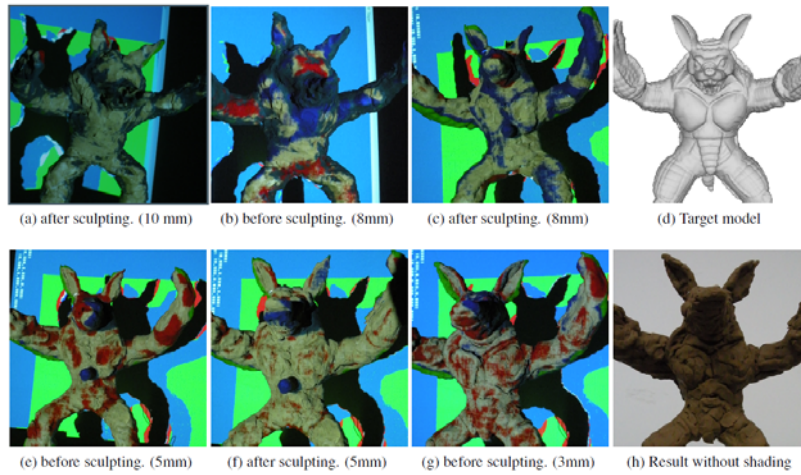
void main(
out float3 outColor : COLOR,
float4 pos : TEXCOORD0,
uniform float threshold,
uniform samplerRECT zbuffer)
{
float depth = ((pos.z/pos.w)+1)/2;
float x = ((pos.x/pos.w)+1)/2;
float y = ((pos.y/pos.w)+1)/2;
float3 p = float3(x*1024,y*768,1);
float t=texRECTproj(zbuffer, p);
float delta =abs(depth-t);
if (delta < threshold) then
    outColor = float3(1,1,1);
else
{
    int level = delta/threshold;
    int n = level-(level/2)*2;
    if (t < depth) then{
        if (depth < 0.01 )then
            outColor = float3(0,0,0);
        else{
            if (n > 0) then
                outColor = float3(1,0,0);
            else
                outColor = float3(1,1,0);}}
    else{
        if (t > 0.99) then
            outColor = float3(0,1,0);
        else{
            if (n > 0) then
                outColor= float3(0,0,1);
            else
                outColor = float3(0,1,1);}}}}
}

```

---

#### Code 1: Pixel shader

Any part of the working model that is out of the silhouette will be rendered as green. To discriminate coarse-to-fine on a different scale, the system provides a dynamic adjustment capability. In the coarse scale, more saturation of the blue or red color is represented as a large difference, indicating that this area should be corrected first. At the fine-tuned scale, most of the differences will not be easy to discriminate. Thus, we can scale down the threshold to a lower value. Then the greatest saturation color can be represented as only one or two millimeters. Figure 7 shows the difference between the coarse and fine rendering scales. The system also provides a selection mechanism that allows the user to select an area of interest to show the exact difference.



**Figure 7: Different stages of the sculpting process. (a,b,c) are in the coarse-tuned stages. (e,f,g) are in the fine-tuned stages. The value in parentheses that represents a difference of less than that value is allowed and shown as white.**

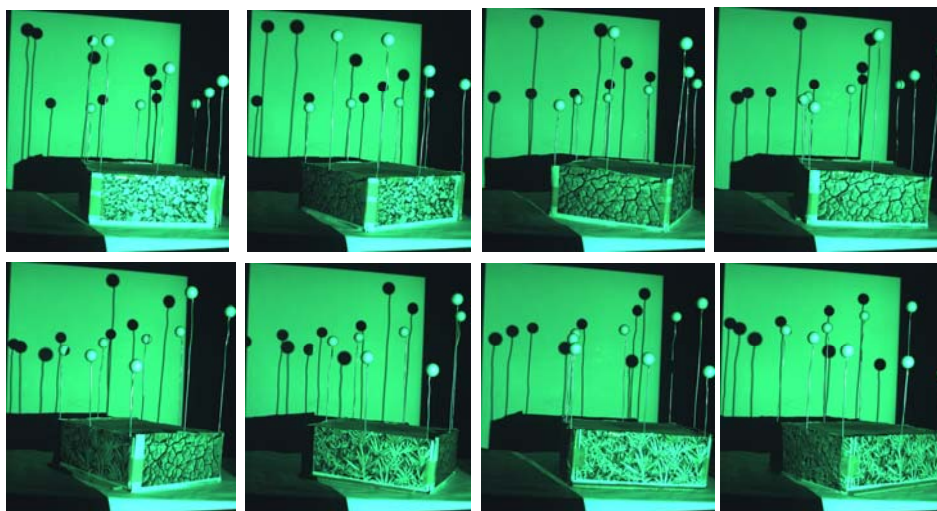
### 5. REGISTRATION

To register two point cloud data sets quickly and robustly is not a trivial problem. Scale-invariant feature transform (SIFT) is applied to two images to get a set of mapping pairs. The Iterative Closest Point (ICP) algorithm is applied to get the transformation. To stabilize the transformation, the desktop plane was estimated first.

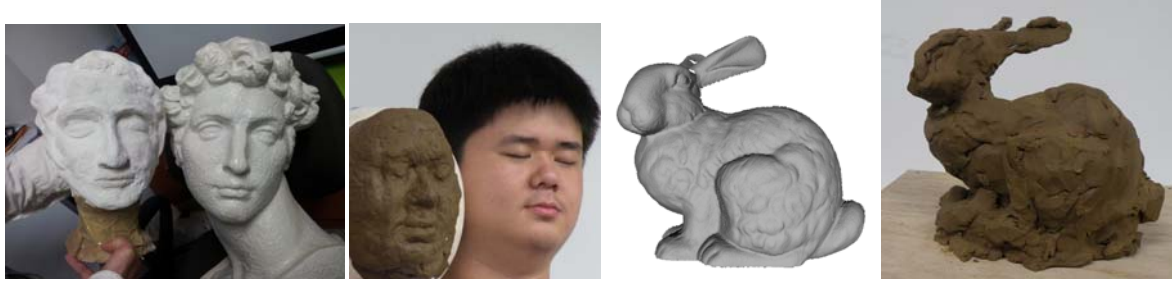
The statue may change position. Thus, the system should detect such an event and make a corresponding transformation. To detect such an event, we should place some fixed points in the scene as Figure 8 shows. Since in the upper area is the working area, we place the reference image in the bottom stage but then we can see only one reference image most of the time. Thus, we should find the twelve reference positions during setup. To find their positions in the initial stage, some markers of different heights are placed in the scene. Each reference image will capture all the markers. Because

each marker is a different height, the system will easily identify the corresponding relationships among these images, and the transformation among images can be recovered. Each time a new scan begins, the system will find the best matched reference image when a new image has been captured and find the transformation from the current position to the reference position.

To increase the robustness of the estimation and reduce the amount of input data, the system will separate the foreground region from the background image. For easy detection of the foreground image, the system will store a background image in advance. When beginning a new scan, the system will detect the region of interest. Then, it will determine the location of the reference image. For rotation and scale invariants, SIFT feature descriptors are detected. To filter out outlier features, the RANdom SAMple Consensus (RANSAC) algorithm is used to find the reference plane. Finally, we can get features exactly



**Figure 8: Estimate the transformation of each reference image at the bottom stage.**



**Figure 9: The comparison of the result with the target model**

on the plane.

To estimate the transformation, the ICP algorithm is an option. Based on the small region extracted from the previous step, we can get more robust matching features. However, there are also some false estimations. To solve this problem, we could limit the rotation to only the desktop.

## 6. RESULT AND FUTURE WORK

Figure 9 show the comparison of the results with the target models. During the sculpting process, we have learned a lot about the user experience, which we will discuss in the following section.

At the beginning of the implementation, the system did not provide the registration feature. Thus, the user cannot change the position of the fabrication or the fabrication and the virtual model will not align on the same basis all the time. With the help of the referenced image the registration problem is solved. Consistent progress is a big issue for users of such a system.

During the sculpture process, users may focus on different types of information. In the initial stage, the user only pays attention to the silhouette of the target model. At the coarse modeling stage, the user is only concerned with the biggest differences. Thus, a varied scale of difference rendering is provided.

For rendering the color difference, paper clay is the best choice. However, paper clay can easily become hard. The moisture needs to be kept inside the clay in order to keep it soft. Otherwise, after hardening, changing the shape of the clay will be more difficult. But if there is too much moisture inside the paper clay, it will become too soft to sculpt its shape. In general, sculpting a statue is a time-consuming process. Even an experienced sculptor may need weeks or months. Oil-based clay may be a better choice because it can remain malleable even when left for long periods in dry environments. However, the color of the oil-based clay is too dark. Not only is it hard to render the color difference, but also it is not easy to scan its 3D shape. To solve this problem, we

change the color of the structured light to green to let its reflection become easier to discriminate.

Art is a creative process. A system, such as presented in this paper, cannot replace the artist's work. However, the system can act as a guide to teach a novice user how to sculpt. With the system's help, even a novice user can construct a prototype in a few hours. However, there are also certain limitations under the current implementation.

The correct depth cannot be estimated in a shadow or a black area. For a human statue, the nose, chin, and hair cannot be modeled well. The color of the modeling material cannot be too dark. Otherwise, the system cannot capture its 3D structure. To reduce the effect of the problem, a suitable model and material should be chose.

The system's behavior is a discrete guide. When a user sculpts the model for a while, he should ask the system to create a new image and show the difference in the current state. The user should stop and wait for the system. The scanning process and computation time is about five seconds. If the user cannot wait, the system can provide automatic scanning and show the differences in a continual mode so that in each modification of the model, the user can see the change immediately. This is more productive than manual scanning. It is important to note that during the sculpting period, the hand of the user is usually dirty. It is not convenient to manipulate the system with a keyboard or mouse. If the system could capture the range data in a continual mode and recognize the position of the hand as a control event, the user would be very happy with this feature.

## Acknowledgements

This work was supported in part by the National Science Council under the grants of NSC 102-2221-E-126 0-008. We also want to acknowledge Tyler Johnson, Herman Towles, Andrei State, Greg Welch, Anselmo Lastra, and Henry Fuchs for discussion and technology support in the prior project. We also acknowledge Sheng Yi Gao, Yu Ren Deng and Wei Heng Shih for testing this system.

## 7. REFERENCES

- [Ban01a] Bandyopadhyay, D., Raskar, R., and Fuchs, H. Dynamic shader lamps: Painting on movable objects. *Augmented Reality*, 2001. Proceedings IEEE and ACM International Symposium on, pp.207-216, 2001.
- [Bha07a] Bhasker, E., Juang, R., and Majumder, A. Registration techniques for using imperfect and partially calibrated devices in planar multiprojector displays. *IEEE Transactions on Visualization and Computer Graphics* 13, No.6, pp.1368–1375, 2007.
- [Bou00a] Bouguet, J.Y. Matlab camera calibration toolbox. 2000.
- [Cho07a] Choi, H., Kyoung, D., and Jung, K. Real-time image correction for interactive environment, 2007.
- [Dao07a] Dao, V.N., Hosoi, K., and Sugimoto, M.. A semi-automatic real-time calibration technique for a handheld projector, 2007.
- [Dra09a] Drareni, J., Roy, S., and Sturm, P. Geometric video projector auto-calibration. *Computer Vision and Pattern Recognition Workshops*, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on, pp.39–46.
- [Fla06a] Flagg, M., and Rehg, J.M. Projector-guided painting. Proceedings of the 19th annual ACM symposium on User interface software and technology, UIST '06, pp.235–244, New York, NY, USA, 2006. ACM.
- [Fla09a] Flagg, M., Skeels, C., and Rehg, J.M. Computational support for creativity via capture and access. 2009.
- [Gri06a] Griesser, A., and Van Gool, L.. Automatic interactive calibration of multi-projector-camera systems. *Computer Vision and Pattern Recognition Workshop*, 2006. CVPRW '06. Conference on, pp.8–8.
- [Ish07a] Ishihara, Y., and Ishihara, M. Locating a projector using the strength of beams reflected on a screen, 2007.
- [Joh06a] Johnson, C.M., Bhat, A., and Thibault, W.C. An evolutionary approach to camera-based projector calibration, 2006.
- [Kah10a] Kahn, S., Wuest, H., Stricker, D., and Fellner, D.W. 3d discrepancy check via augmented reality. In *Mixed and Augmented Reality (ISMAR)*, 2010 9th IEEE International Symposium on, pp.241–242, 2010.
- [Kim07a] Kimura, M., Mochimaru, M., and Kanade, T. Projector calibration using arbitrary planes and calibrated camera. In *Computer Vision and Pattern Recognition*, 2007. CVPR '07. IEEE Conference on, pp.1–2.
- [Lan09a] Lanman, D., and Taubin, G. Build your own 3d scanner: Optical triangulation for beginners @ONLINE, 2009.
- [Löc11a] Löchtefeld, M., Gehring, S., Jung, R., and Krüger, A. guitAR: supporting guitar learning through mobile projection. *CHI'11 Extended Abstracts on Human Factors in Computing Systems*, ACM, pp.1447–1452, 2011.
- [Oka09a] Okatani, T., and Deguchi, K. Easy calibration of a multi-projector display system. *Int. J. Comput. Vision*, 85, No.1, pp.1–18, 2009.
- [Ras01a] Raskar, R., Welch, G., Low, K., and Bandyopadhyay, B. Shader lamps: Animating real objects with image-based illumination. In *Rendering Techniques 2001, The Eurographics Workshop on Rendering*, pp. 89–102, 2001.
- [Ras03a] Raskar, R., van Baar, J., Beardsley, P., Willwacher, T., Rao, S., and Forlines, C. ilamps: geometrically aware and self-configuring projectors. *ACM Trans. Graph.* 22, No.3, pp.809–818, 2003.
- [Riv12a] Rivers, A., Adams, A., and Durand, F. Sculpting by numbers. *ACM Trans. Graph.* 31, No.6, pp.157:1–7, November 2012.
- [Saj09a] Sajadi, B., Lazarov, M., Gopi, M., and Majumder, A. Color seamlessness in multi-projector displays using constrained gamut morphing. *Visualization and Computer Graphics*, IEEE Transactions on, 15, No.6, pp.1317–1326, 2009.
- [Saj09b] Sajadi, B. and Majumder, A. Marker less view-independent registration of multiple distorted projectors on extruded surfaces using an uncalibrated camera. *Ieee Transactions on Visualization and Computer Graphics* 15, No.6, pp.1307–1316, 2009.
- [Saj10a] Sajadi, B., and Majumder, A. Auto-calibration of cylindrical multi-projector systems. In *Virtual Reality Conference (VR)*, IEEE, pages 155–162, 2010.
- [Saj10b] Sajadi, B., and Majumder, A. Scalable multi-view registration for multi-projector displays on vertically extruded surfaces. *Computer Graphics Forum* 29, No.3, pp.1063–1072, 2010.
- [Sun08a] Sun, W., Yang, X., Xiao, S., and Hu, W. Robust checkerboard recognition for efficient nonplanar geometry registration in projector-camera systems, 2008.
- [Zha14a] Zhang, Y., Xiong, Z., Yang, Z., and Wu, F.. Real-time scalable depth sensing with hybrid structured light illumination. *Image Processing*, IEEE Transactions on, 23, No.1, pp.97–109, 2014.
- [Zho08a] Zhou, J., Wang, L., Akbarzadeh, A., and Yang, R. Multi-projector display with continuous self-calibration, *Proceedings of the 5<sup>th</sup> ACM/IEEE International Workshop on Projector camera systems*, 2008