3D Scanning and Modeling System Based on FPGA

Yu-Ning Jiang[1], Zi-Hong Su, and Shui-Sheng Xiao

Department of Microelectronics, School of Physics and Engineering, Sun Yat-Sen University,
Haizhu District, Guangzhou, Guangdong, 510275, China

[1]285551269@qq.com

Abstract— ***3D scanning is an important technology for today's 3D applications, especially when a universal model of a real-world object is needed. Laser scanning techniques are developed, but it is difficult for it to be common used because of their complexity and high costs. A 3D scanning and modeling system with non-contact passive techniques is presented in this paper. The system is based on Altera's Cyclone II FPGA, and by using the SOPC tools, components related to the system are integrated on the FPGA chip. We came up with image processing and modeling algorithms that are very suitable for FPGA, and made a simulation in MATLAB, the algorithms turn out to be fast and robust. And the paper shows the concrete architecture of our design. The system is cheap, simple in structure, highly integrated, relatively independent, and can make 3D models in .STL form.***

Keywords — **3D; scanning; modeling; FPGA; MATLAB; SOPC; image processing**

I. Introduction

With the fast development of 3D printer and other 3D applications, 3D scanning, as a good approach to construct three-dimensional models, is now attaching more and more attention. 3D scanning technology and the correlative devices are developing on the orientation of high accuracy, high efficiency and low cost.

The common function of 3D scanners is, to analyse a real-world object or environment to collect data on its shape and appearance. The collected data can then be used to construct digital three dimensional models. There are many kinds of 3D scanners with different methods, and a well established classification divides them into two types: contact and non-contact 3D scanners [1]. We will focus on the latter one in this paper. Non-contact 3D scanners include non-contact active and non-contact passive ones. Non-contact active scanners are in need of laser or other special light sources, they emit some kind of radiation or light and detect its reflection or radiation passing through object in order to probe an object or environment [2]. On the contrary, Non-contact passive scanners do not emit any kind of radiation themselves, instead it relies on detecting reflected ambient radiation. Passive scanners utilize visible light so the systems are generally simpler than Active scanners, their techniques mainly include stereoscopic techniques, photometric techniques and silhouette techniques.

Silhouette techniques are almost the simplest. They use outlines created from a sequence of photographs around a three-dimensional object against a well contrasted background [2] to obtain multi-view photographs of the visual hull, a series of cameras are usually required. Besides, a cheaper way is to build a rotary platform, and use only one camera to take the side views, like Figure 1. With this method, we designed a 3D scanning and modeling system.

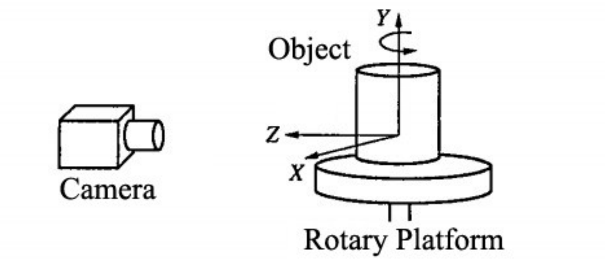

Figure 1. A 3D scanner with a rotary platform and a camera [3]

What we tried to make was a whole system, rather than just a simple scanner. The system would implement the following work: rotary platform controlling, image capturing, image processing, modeling(three-dimensional reconstruction) and file output. The file would be output within binary .STL format (Stereo Lithography), which is a type of widely used 3D format and can be imported by 3D processing softwares or 3D printers. We are going to implement our system on the Altera DE2-70 FPGA.

An essential drawback of silhouette techniques is that some concavities of an object cannot be detected, so it's hard for these approaches to reconstruct complicated 3D models. And compared with some laser scanning techniques, silhouette techniques are less accurate. Yet, in spite of this, our scanning and modeling system has its own merits. Firstly, the system is simple, what we need are just an ordinary digital camera, a rotary platform and several controlling and processing modules. Secondly, the system is much cheaper than laser scanners, the price of most laser scanners on sale is over \$3000, while the price of our system will be about 10% of it, according to our evaluation. Thirdly, the system is safe, without laser the system does no harm to the human body or the objects.

The most important feature of our system is that it is based on FPGA. Most of the non-contact passive scanners transmit image data to PC and rely on the software on PC to process the data and reconstruct 3D models, this method occupies much resource of PC and may affect other work on PC. Our system is independent. We develop algorithms that are quite suitable for FPGA, so FPGA can undertake almost all the processing and modeling work itself, and finally transfers finished files to PC via USB. Our goal is to integrate the processing modules on a chip so we make full use of the flexibility and high efficiency of FPGA design. We can complete most of the processes with the sufficient logic resources on the developing board so the speed will be fast, and with the help of the strong and efficient ability of data process of the Nios II CPU we can achieve some advanced functions.

This paper will introduce our methods and algorithms, and show the simulation results of our algorithms. We tested out our algorithms in MATLAB and the results were satisfactory. Then this paper will show the structure of our design. In the end, we will give a brief conclusion to our design.

II. Image Processing and Modeling Algorithms

A. Multi-View Image Capturing

We need a digital camera and a rotary platform to achieve the function of multi-view image capturing. The platform is drived by a stepping motor, and the motor is controlled by our FPGA, which integrates a stepping motor IP core [4]. The stepping angle of the motor is 1.8° , so the motor runs 200 steps in each circle. After each step the camera captures an image and stores it in the SDRAM, when the image processing is over, the image will be deleted.

B. Image Processing

The purpose of image processing is to acquire the silhouette of the object. The concrete operations include background recognition, image binarization and denoising.

We use an adaptive threshold algorithm to recognize the background color [5][6]. Our algorithm is simple, because there is no need to detect the edge. We can place a pure colored screen behind the object as its background to make it easier to be distinguished. We take sample from the image and note down its RGB brightness value, then we will obtain the RGB brightness frequency distribution. And we assign different weight to the pixels, if the position of the pixel is more likely to be background, it will be given more weight, as is shown in Figure 2.

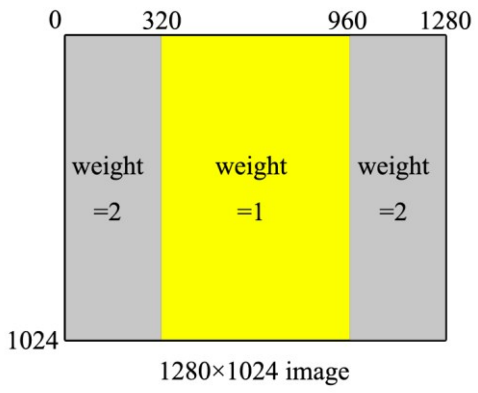

Figure 2. The weight distribution in a 1280×1024 image

According to the RGB brightness frequency distribution we can define the range of RGB brightness for the background color. The algorithm will detect the frequency peak, and work out the upper limit and lower limit (threshold), where the frequency of RGB brightness is 1% of the peak value. For instance, Figure 3 is the blue brightness frequency distribution of row 509~512. The peak is prominent and we obtain the blue brightness range of the background color is 194~203.

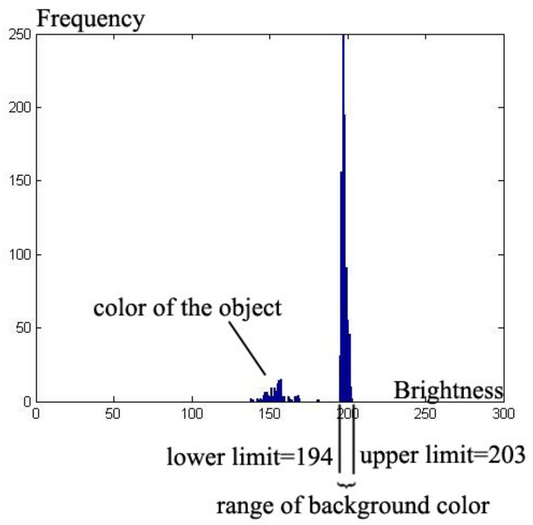

Figure 3. The blue brightness frequency distribution of row 509~512

Actually, the background color in an image is not always the same, so for every four rows of pixels we define three ranges for red, green and blue brightness. We don't use the grey scale map because it loses some important color information.

In MATLAB, this adaptive threshold algorithm takes 7.5660 seconds on average. The algorithm is executed when the first image is captured, and it will be executed only once. Once the threshold values are obtained, the system can execute the image binarization algorithm directly, then it will be much faster. For each pixel, the binarization algorithm judges whether its RGB brightness is in the range or not, if it's RGB brightness is in the range, the pixel will be regard as a background pixel and will be assigned "0", else it will be regard as an object pixel and will be assigned "1". In MATLAB, the binarization algorithm takes 0.4836 second on average. The result of binarization is shown in Figure 4.

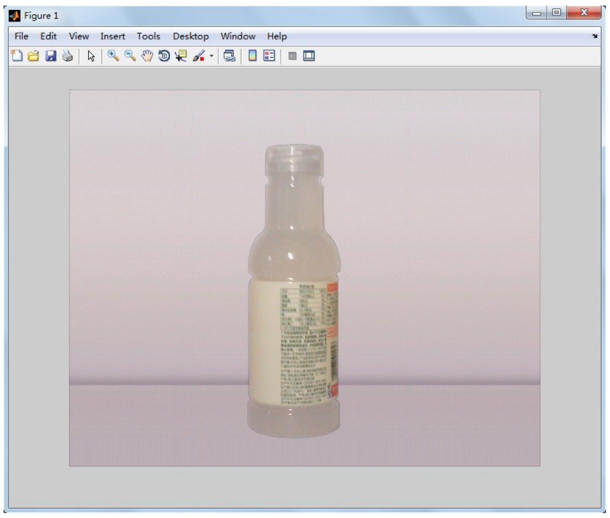

Figure 4(a). Original image

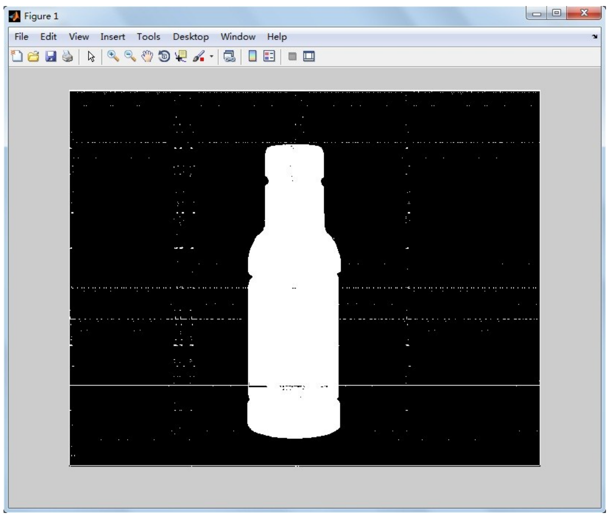

Figure 4(b). After binarization

The result of binarization is satisfactory, but the noise is unavoidable. So a denoising operation is needed. We execute a Morphological Closing operation with matrix A to remove the noise in the object, and then execute a Morphological Opening operation with matrix B to remove the noise in the background [7].

$$A=\begin{bmatrix} 0&0&1&0&0\\ 0&0&1&0&0\\ 1&1&1&1&1\\ 0&0&1&0&0\\ 0&0&1&0&0 \end{bmatrix} \tag{1}$$

$$B=\begin{bmatrix} 0&0&0&1&0&0&0\\ 0&0&0&1&0&0&0\\ 0&0&0&1&0&0&0\\ 1&1&1&1&1&1&1\\ 0&0&0&1&0&0&0\\ 0&0&0&1&0&0&0\\ 0&0&0&1&0&0&0 \end{bmatrix} \tag{2}$$

After the denoising operation, the image becomes clear, as is shown in Figure 5. The operation takes 0.7020 second on average. And the operation can be easily implemented on FPGA.

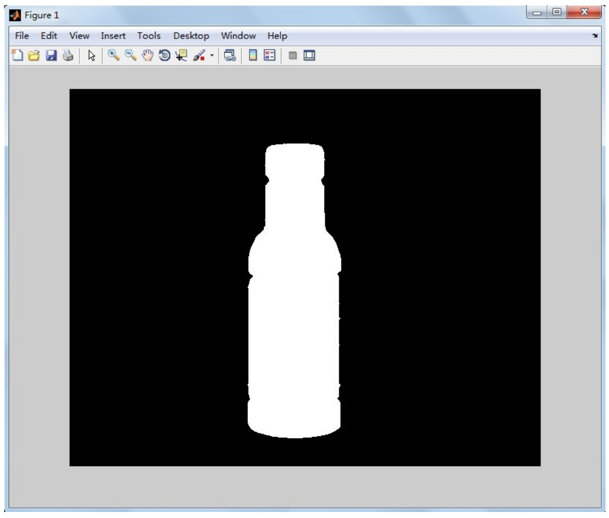

Figure 5. The result of the denoising operation

C. Axis Calibration

In our silhouette method, we must measure the position of the rotary axis. This operation is important, because it ensures the correct shape of our 3D model, we will discuss it later. We execute an "or" logic operation between pixels from two images, the 0° and 180° image. And then we take several rows as sample, and work out the horizontal coordinates of their center of gravity, then work out the median value as the position of the rotary axis. This algorithm takes 0.1092 second on average, and it will also be executed only once.

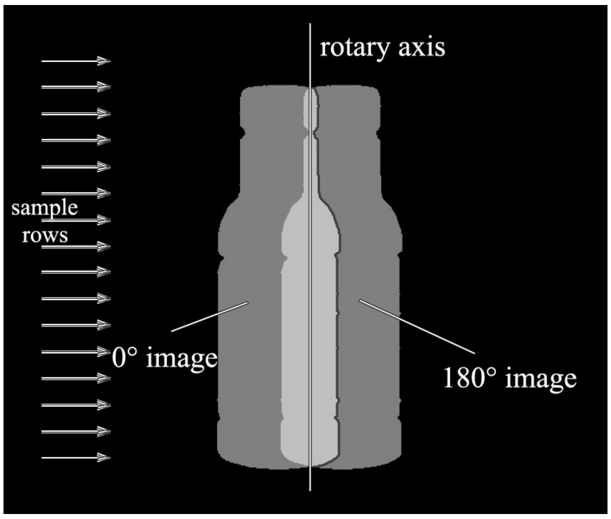

Figure 6. The schematic diagram of axis calibration

D. Modeling

The most important part of our algorithms is the modeling algorithm. Here the definition of 3D modeling is using the silhouette images to reconstruct a virtual, digital three-dimensional model, and the model is essentially a series of point clouds.

The main principle of 3D modeling with silhouettes is shown in Figure 7. After image processing we obtain the silhouettes of the object. As we have mentioned above, the stepping motor runs 200 steps in each circle, so we have 200 viewpoints, for each viewpoint we obtain an image of the silhouette. Now imagine the silhouettes are projected to a virtual space, and shape of their intersection will be like the object. In other words, we can compare the projection to photo lithography, the silhouettes are like the masks, if any parts of the virtual model are not shielded by the silhouettes, these parts will be etched away. The more silhouettes we use, the more accurate the model will be.

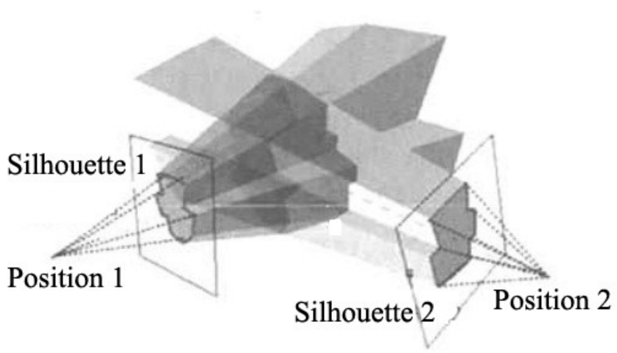

Figure 7. The schematic diagram of 3D modeling with silhouettes [8]

The system captures silhouette images one by one, on PC we can store them all and can select any of them to continue the next process, but on FPGA, we would like to save the memory and improve the speed. Therefore, each image will not be stored for long, the system calls it once, and after some processes it will be deleted. What the system should store is the intermediate product of modeling. Some old methods store all the points inside a model, then gradually reduce the points, until all the silhouettes are used. These old methods require a large memory and numerous operations. Some methods are more advanced, they store the joints of the intersection [9], but they have to solve numerous equation sets to acquire the coordinates of these joints, which involves too many floating-point calculations, so these methods are not suitable for FPGA.

We have come up with a new method of modeling that is suitable for FPGA. We project the silhouette to a cylindrical coordinate system, rather than a three-dimensional Cartesian coordinate system, like Figure 8. As is discussed above, an axis carlibration operation is taken, so if we align the rotary axis at the z axis, the cylindrical coordinate system will be exactly corresponding to the real-world rotary platform. Then the radial distance in the cylindrical coordinate system ('r') will be corresponding to the horizontal coordinate in the image ('x'), and the height ('z') will be corresponding to the vertical coordinate in the image ('y'), and the angular coordinate ('Φ') will be corresponding to the rotation angle of the platform (also means the number of the image, the step of the motor, or the run time).

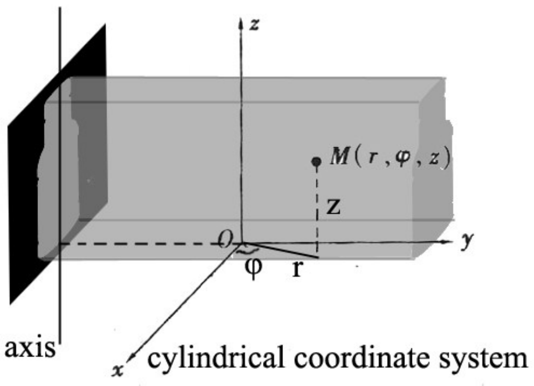

Figure 8. The schematic diagram of the projection in cylindrical coordinate system

After alignment, there is no need to solve any equation sets. The system doesn't work out the intersection among projections, but works out the intersection between a projection and the virtual model in the cylindrical coordinate system. And after working out this intersection, the system doesn't store all the points, but several groups of boundary points, which include starting points and ending points. The schematic diagram is drawn in Figure 9, the system is executing a projecting operation for a row of pixels in the image (and for each row the system will do the same thing). It's a top view, P1 is the starting point and P2 is the ending point, they are signed in the previous projecting operations. From P1 to P2, the system makes a projection of each point on the axis Φ, if the point is corresponding to "1" (silhouette), the point will remain; if the point is corresponding to "0" (background), the point will be deleted. For instance, in Figure 9, P1 will be deleted and P2 will remain. We set up a sign to note down whether the point is corresponding to "1" or not, once the sign changes, note down the coordinate of the point, it will be a new starting point or ending point, like P3. So after a series of comparing operation, several groups of new starting points and ending points are noted and stored, that means the model is refreshed.

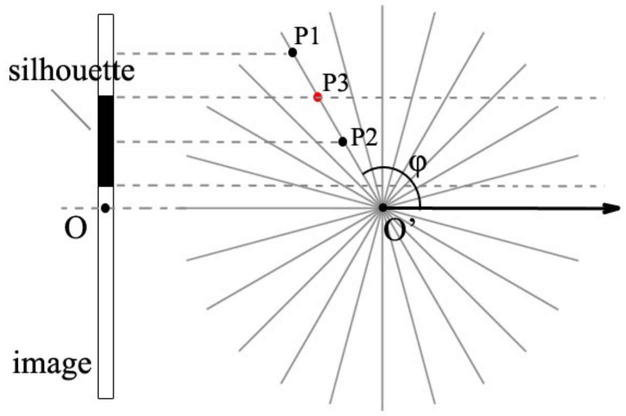

Figure 9. A top view of the projection in cylindrical coordinate system

Since the system only stores the coordinates of starting and ending points, it costs less memory. Theoretically, the memory that the work place occupies is lower than 4 MB, as well as the model data. So the system can store the intermediate data in SDRAM.

This type of algorithm is fast, because it requires fewer floating-point calculations, and cleverly avoid the operation of working out the intersection among projections, and stores less data. We have tested out that in MATLAB the whole modeling algorithm takes 10.1245 seconds, and it produces about 200,000 point clouds.

E. File generation and output

After the modeling operation, the system stores all the data of the visual hull of the object, but the data is not the final file, it should be transformed into a proper format. The system will transform cylindrical coordinates into three-dimensional Cartesian coordinates, and then execute a triangularization operation. The triangularization operation is simple. For every two adjacent angles and two adjacent height coordinates, there is a facet. Each facet has four vertexes, and can make two triangles. In the .STL format, what the system should write are the coordinates of vertexes in each triangles. And the length unit is "pixels", if we add a scaling operation, it can be transformed into "centimetre" or other units.

The final three dimensional model is shown in Figure 10, it turns out to be perfect. So in MATLAB we have simulated our algorithms, which turns out to be fast and robust, then we will try to implement the system on FPGA.

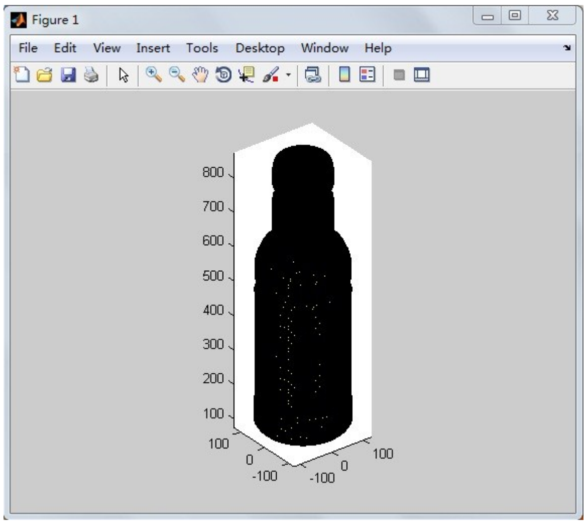

Figure 10. A perspective of the 3D model

III. System Architecture

A. General framework

The system architecture is shown in Figure 11.

In our design, we use Cyclone II FPGA, and take advantage of the resources of the FPGA develop platform and DE2-70 develop board. The hardware system of the design is built and each functional module is designed by Verilog HDL on Quartus II. By using the SOPC develop tools, a NIOS II CPU, memory devices, I/O interfaces and other components related to the system are integrated on the FPGA chip. The data of the system are under control of Avalon bus. We design several hardware modules to complete most of the image processing operations and part of the modeling operations, so the efficiency is high. And the software system of the design is developed, the device drivers of the components of the system and data process are accomplished on Nios II EDS. And the design benefits from IP cores provided by Altera. Cooperating hardware design with software design, the design of a highly integrated 3D scanning and modeling system will take advantage of both the sufficient resources on the developing board and the strong and efficient ability of data process of the Nios II CPU.

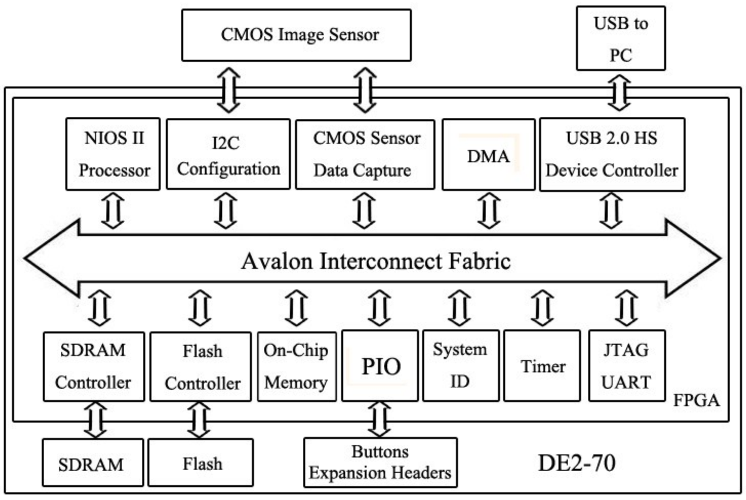

Figure 11. The system architecture of this design

B. Specification

1) Rotary platform: The rotary platform is an important peripheral device, which should bear the weight of the object and show the whole visual hull. To build the platform, we need a turntable and an accurate stepping motor. The motor is controlled by our FPGA via the expansion headers. The stepping angle of the motor is 1.8°, so the motor runs 200 steps in each circle. 200 steps can make sure the system captures sufficient silhouette data. Besides, the horizontal of the rotary platform is crucial.

2) Digital camera: In the multi-view image capturing, a TRDB-D5M COMS sensor is required. The COMS sensor is controlled by I2C bus, and captures 1280×1024 RGB images. The accuracy is sufficient.

3) Image processing modules: The image processing modules are mainly achieved with logical circuits, which are described by Verilog HDL. The processing methods have been introduced in the algorithm section above. What's more, some buffer devices are required.

4) Memory devices: There are several types of memory devices we need. On DE2-70 development board there are two SDRAMs, totally 64 MB, they are used as image buffer and the modeling work place. The Flash is used to store our program and configuration informations. The on-chip memory is small but fast, we use it as a buffer device when the system is operating. No SD card is needed, for the access of an SD card is too slow.

5) Modeling and file generation modules: What the modeling models should do is mainly comparing and accessing repeatedly, so it is not difficult to implement on FPGA. Some sine and cosine values are required, we should define these values in advance, so FPGA doesn't need to do the sine and cosine operations itself. After modeling, all images are deleted, and the file generation modules begin to work. They are designed in accordance with the algorithms above.

6) JTAG UART and USB: Although our system is relatively independent, it has to communicate with PC at least twice: program downloading and file transmission. JTAG UART and USB will implement these functions.

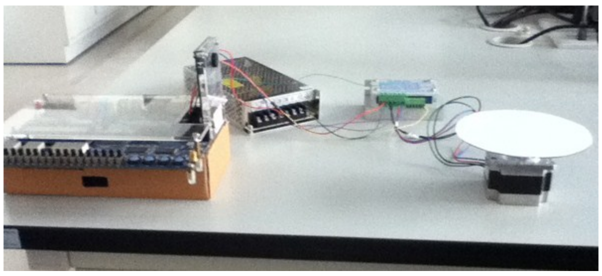

Figure 11. The appearance of our preliminary system

7) NIOS II CPU: The NIOS II CPU is the centre of the system, which controls the operation of the whole system, such as data transmission and module drive. Besides, it runs the soft IP cores and deals with some relatively difficult operations.

With the modules and devices mentioned above, we have formed a preliminary system. The system appearance is shown in Figure 12. We are now working on some unfinished code and adjusting some parameters, the result of the system performance will be seen in the near future.

IV. Conclusions

In this paper, a 3D scanning and modeling system based on FPGA is introduced. The system would implement the following works: rotary platform controlling, image capture, image process modeling and file output.

The main principle of our design is the silhouette method, we capture multi-view images with a digital camera and a rotary platform, which saves the cost and also collects sufficient silhouette data.

We come up with some algorithms that are suitable for FPGA, and we have made a simulation in MATLAB. We use an adaptive threshold algorithm to recognize the background color, the algorithm is sensitive and there is no need to detect the edge. And then we execute the image binarization operation and denoising, the results turn out to be perfect.

In the modeling operation, our algorithm is much easier than others. We project the silhouette to a cylindrical coordinate system, rather than a three-dimensional Cartesian coordinate system. The algorithm doesn't need to solve any equation sets or do numerous floating-point operations, what it actually does is comparing and accessing, so the algorithm is fast and suitable for FPGA. After modeling, the system can output 3D model files in .STL format.

Then the system architecture is introduced. Our design uses Cyclone II FPGA, and takes advantage of both the sufficient resources on the developing board and the strong and efficient ability of data process of the Nios II CPU. The system is cheap, simple in structure, highly integrated, and relatively independent.

An essential drawback of silhouette techniques is that some concavities of an object cannot be detected, but when modeling for normal convex objects, our system is useful and has its own advantages, and compared with laser scanners, the system is safe and does no harm to the human body or the objects.

In future,, we are planning to make further improvement for our system, especially the Verilog codes of the modeling and file generating modules. And we are going to improve our algorithms, they can be faster and simpler.

3D scanning is a field with great promise, and many new methods and new products are developing. 3D scanning techniques and the correlative devices are developing on the orientation of high accuracy, high efficiency and low cost. And we believe that the 3D scanning and modeling techniques, or even machine vision will become much smarter in the future, which will change our life and work style greatly.

Acknowledgment

The research is supported by *Laboratory of FPD & System Integration Postgraduate Education Practice Base*. Special thanks to our advisors, Shaozhi Deng and Zhiyong Pang.

References

[1] Raymond A. Morano, Cengizhan Ozturk, Robert Conn, Stephen Dubin, Stanley Zietz and Jonathan Nissanov, "Structured Light Using Pseudorandom Codes", IEEE Transactions on Pattern Analysis and Machine Intelligence - TPAMI, vol. 20, no. 3, pp. 322－327, 1998

[2] The Wikipedia website, "3D Scanner" [Online]. Available: http://en.wikipedia.org/wiki/3D_scanner.

[3] Gao Zhu, Ji Xiaomin, Wang Fang and Xue Mei, "A Method for 3D Reconstruction of a Product's Shape Based on its Photos", Mechanical Science and Technology for Aerospace Engineering, Vol.28, No.5, May. 2009.

[4] WANG Bang-ji, LIU Qing-xiang, ZHOU Lei, LI Xiang-qiang, ZHANG Jian-qiong, "FPGA-based multiple-axis stepper motor controller",Electric machines and control, Vol.16, No.3, Mar.2012.

[5] Wen Pinggeng, Liu Chun and Li Jian, "A Smart Image Background Recognition Model and its Application to Content-based Image Retrieval", Intelligent Systems and Application (ISA), ISBN 978-1-4244-5874-5, May. 2010.

[6] Jihong Liu and Chengyuan Wang, "An Algorithm for Image Binarization Based on Adaptive Threshold", Control and Decision Conference, 2009. ISBN 978-1-4244-2723-9, June 2009.

[7] R. C. Gonzalez, R. E. Woods and S. L. Eddins, "Digital Image Processing Using MATLAB", ISBN 7-121-01456-4, Sept. 2005.

[8] Zhang Hon, Xiong Hanwei, Wang Jinming and Zhang Xiangwei, "Scanning Freeform Objects by Combining Shape from Silhouette and Shape from Line Structured Light", Applied Laser, Vol.28, No.2, April 2008.

[9] NGUYEN Manh-quy and ZHANG Yu-jin, "Fast 3D model generation from silhouettes", Joumal of Computer Applications, Vol. 30, No. 11, Nov. 2010.