

Low-cost and effective Structure-from-Motion 3-dimensional geometric reconstructions of targets at macroscopic scale

1

Timothy Bauch *Undergraduate, Imaging Science, Rochester Institute of Technology, December 2016*

Abstract— When an artist creates a painting there are always concerns over being able to properly protect the authenticity of their work. Counterfeiters may try to replicate the artwork and claim it as genuine. If an artist can accurately ascertain the surface topography of their artwork, they will be able to win any legal dispute of the authenticity of such work. There are currently many methods with which they may be able to properly image and protect their artwork. High-resolution, topographic detailing usually carries a high cost and may not be economically feasible for the average artist. This paper proposes a low cost, portable, and easy to use system that an artist, or other practitioner, may be able to use in order to measure surface structure on their paintings, currency, or other documents in order to protect its authenticity. Through the combination of a computerized numerical controlled (CNC) machine with a common DSLR camera we have been able to put together a low-cost system that will give us all the data necessary for 3D surface reconstruction. This data can then be sent into any structure-from-motion photogrammetry software program on the market and produce an accurate model of the surface at a fraction of the cost of many other high-end surface-topography systems. This device has shown promise as a prototype off the shelf imaging system with a high degree of accuracy for the relatively low price point of the system.

Index Terms— Structure-from-Motion, topographic, CNC, reconstruction, 3-dimensional, macro, photography, photogrammetric, accuracy, high-resolution, paintings, surface, measurements

I. INTRODUCTION

SURFACE topography has been a very popular source of research for many years. So why is surface topography so important? This question can be answered when discussing three constituent characteristics of the surface texture on any object; lay, surface roughness, and waviness.[14]

One might be interested in the lay of an object such as a painting or handwritten document. The lay of a surface will tell you the surface pattern that was used when the object was created. We would be able to tell the brush stroke a painter

This paper was written and submitted for review by Timothy Bauch in December 2016. The work was conducted with support from the Imaging Science Department at Rochester Institute of Technology. Specifically, support was given by Carl Salvaggio, Roger Easton and Jim Ferwerda.

Timothy Bauch is with the Rochester Institute of Technology Imaging Science Department, Rochester, NY 14624 USA (e-mail: tdb8904@rit.edu).

used, or if a document was written in a certain direction in order to decipher if the creator was left or right handed.

The surface roughness can give us information about what was used to make the object. These would be considered closely spaced irregularities. This can be important as the ability for us to know what was used to create the surface could help us reconstruct another one. A machine tool would leave cutting marks based on the micro profile of the tool or a brush would leave a certain roughness based on the size of the brush head.

Lastly, the waviness of a surface characterizes vibrations or chatter irregularities that would occur during the life of an object. One could be able to determine if the object was warped or deformed over time.

These are just a few reasons one might want to know the surface topography. There are many different techniques that are used to reconstruct the surface topography of an object, however, they also carry a high price tag to produce a working system. For instance, if you were to purchase a profilometer, which is highly accurate, you will most likely spend at least \$50,000. This can be out of the price range for many people who wish to get topographic data. A very common and successful approach to achieve surface topography is structure-from-motion (SfM). This method can develop accurate data while reducing the overhead of expensive equipment that a profilometer uses.

A. Photogrammetric Methods

Over the last 15-20 years there has been a technological boom in the area of computer vision-based photogrammetry.[3] Photogrammetry is the use of photography in surveying and mapping to measure distances between objects. Triangulation is the main principle behind most photogrammetry methods to create a 3-dimensional point measurement from 2 dimensional photos. By mathematically intersecting converging lines in space, you can get a near exact location of the point in 3D space. This is known as epipolar geometry, shown in Figure 1 below.[15]

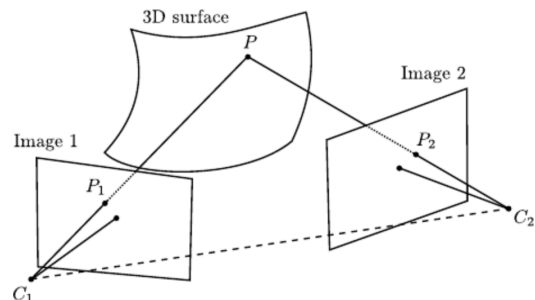


Fig 1. Illustrates the concept in how the reconstruction of a 3D point may be achieved using epipolar geometry [15]

The development of triangulation and image extraction algorithms have enhanced the quality of photogrammetric reconstruction methods. Some methods that have arisen over the last 20 years are stereo computer vision reconstruction, structure-from-shading, structure-from-focus, structure-from-texture, structure-from-motion, structured-light, light detection and ranging (LIDAR) and time-of-flight cameras. Each of these methods acquire range data using different techniques with both benefits and challenges. For the research conducted in this paper, we chose to focus on the technique that gave us the ability to have both low cost and high accuracy in results. Specifically, we looked at those methods that required one camera, simple lighting, ability for close range imaging, and ease of movements. Structure-from-motion was the method we chose to use for this project. We wanted to use a method that afforded the flexibility to use inexpensive parts to capture the data needed. Structure-from-motion gave us the ability to build an affordable system that someone, such as an artist, might be able to afford.

B. Structure from Motion (SfM)

Humans can process a lot of information in the world around us just by moving around. As we move around an object, we are able to process every vantage point and create a model of the object in 3D. When a human looks at an object from one vantage point, we are unable to determine the depth of the object without moving around to see different vantage points. If we want to use a computer to do this exact process, we need to model the same approach humans use. This is where the technique structure-from-motion (SfM) comes into play. Computers are now able to model this 3D reproduction because of different techniques like SfM. SfM is a computer-vision based photogrammetry process that can estimate 3D structures from 2D images.[1] The collection of images that are to be used in a topological estimator are shown in Figure 2.

Structure-from-motion is concerned with the problem of measuring correspondence between images and subsequently conducting the reconstruction of 3D objects.[16] The workflow for SfM can be seen in Figure 3. You start with many images of a scene at different vantage points. In order to create a large enough dataset of images, there needs to be an overlapping of between 80-85 percent in both the fore and side orientations, for optimal reconstruction.

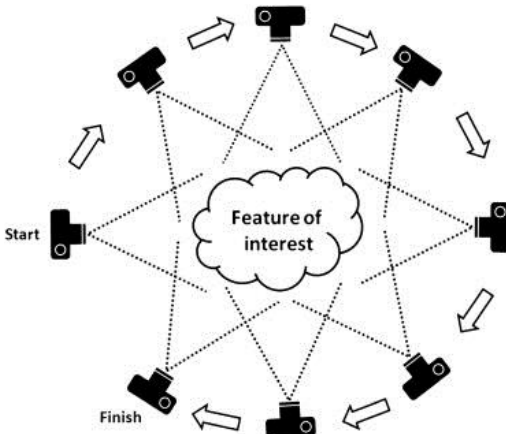


Fig 2. The SfM technique requires multiple images with a high percentage of overlap for input.

Once you have the images you must extract the features from all images. The most common method to do this is the scale invariant feature transform (SIFT) algorithm.[3] The reason this method is used is because of its ability to provide a robust set of feature descriptors between images that might have different conditions. All like features, or descriptors, are then matched together and the correspondences between the images are computed. RANSAC is the most common algorithm to conduct this correspondence between images because of its relatively robust performance in the presence of noise.[3] A point cloud is created from this data and the feature trajectories from the point cloud are looked at to create a distance, or depth, at each point. After the surface reconstruction is generated you can then visually see the final output of the 3D model.

There are currently several leading software programs on the market that use structure-from-motion as the technique for reconstruction. For this paper, Pix4D was used as the modeling software from the images created using our proposed system.[16] This program is known as one of the top algorithms created to carryout SfM, among other visualization tasks. This software has a license fee of \$8,700, however, there are many other software programs on the market for much less. For instance, Agisoft Photoscan offers a stand-alone license for only \$179 and a professional edition for \$3,499. This price was not included in the total cost of our system as there are several products on the market that could be used to process the images from the proposed system.

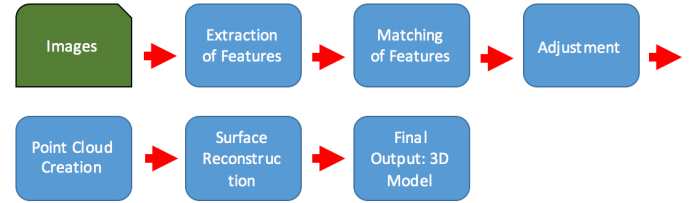


Fig 3. Diagram that features the process chain in the Structure from Motion reproduction method.

Structure-from-motion reconstruction has been under question for its accuracy by researchers.[2] By using accurate ground-truth measurements, researchers have begun to scrutinize the actual ability of any SfM algorithm to model a given scene. It is therefore difficult to accurately ascertain the error within a scene due to the nonlinear statistical nature of the reconstruction.[3] These sources of error will be discussed later in the paper.

II. BACKGROUND

Being able to produce a relatively inexpensive system can be quite challenging when combining the different processes necessary to automate the structure-from-motion process. The first essential element is a camera that can produce high-resolution images. The camera would most likely require a macro lens which is able to resolve fine detail of surfaces for close range targets. The next element is a machine that is capable of moving in a controlled manner through Cartesian spaces. Having these two components the last component would be developing software for a user to

control both of the other components with relative ease. The main goal therefore, was to be able to achieve these subtasks while maintaining an affordable cost for an average user such as an artist.

A. Computer Numeric Control (CNC) Machine

Computer numeric control (CNC) machines are used in manufacturing and have the ability to use computers to control the movement. Tools can be mounted onto the machine and controlled with simple commands. These machines can cost a significant amount of money due to their abilities to be highly accurate and complex in motion.

The goal of the research was to limit the cost of the system, so finding an affordable CNC machine was crucial. The Probotix Fireball V90 (shown below in Figure 4)[17] was the perfect solution to this problem. This machine can be found for \$1,500 and is known to be highly accurate. It has the ability to move in the X, Y and Z directions, which are the movements we are interested in for the system.

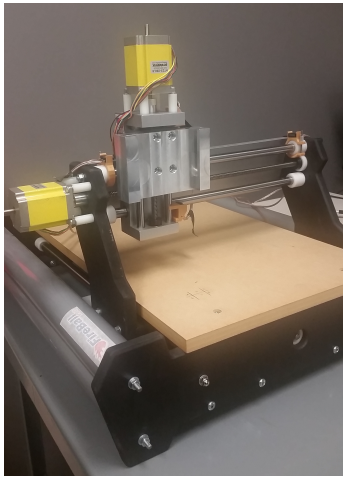


Fig 4. Probotix Fireball V90 CNC machine with stepper motor controls. Originally designed for mounting a router tool, it has the ability to mount different tools for differing uses.

B. DSLR Camera

A DSLR, or digital single-lens reflex camera, is a system that combines the optics and mechanisms of a single-lens reflex camera with a digital sensor instead of film.

In the 1990s, DSLR cameras being made had a sensor with only 2.74 megapixels. Today, most DSLR cameras carry a sensor with around 18-36 megapixels.[18] This change in sensors has given more power to the average camera.

The Canon T2i was chosen for this system as it offers an affordable price tag of around \$300-\$400, which is well in reason for an average person to use. The Canon T2i, also, offers an 18-megapixel sensor with a frame rate from 3.4 to 3.7 frames per second. This camera offers ISO ranges from 100 to 6,400, with a special option of 12,800. An important aspect to consider for this camera is the sensor size. The sensor used is 22.3mm x 14.9mm. Due to the size of the sensor all movements of the camera in the X direction will have to be different than the Y direction to achieve proper 80-85% overlap. Canon no longer produces the T2i, however, there are many comparable cameras that one could find that

will work for this system. The proposed system can take almost any DSLR camera and be able to mount it onto the CNC machine. This is discussed later in this paper.



Fig 5. A Canon t2i DSLR camera with a macro lens and illuminator ring attached. Combined, these have the ability to produce highly resolved photographs that will be needed for input into the structure from motion algorithm.

C. Macro Photography

Macro photography is close-up photography. Macro photography is usually conducted by imaging very small objects and giving it a larger than life appearance in the produced image. This is achieved by what is known as the reproduction ratio. This ratio determines the amount of magnification that the object will have at the sensor. In macro photography the reproduction ratio is typically 1:4.[11]

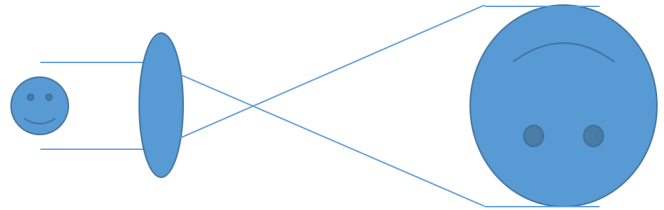


Fig 6. In macro photography the resulting image is always magnified and with great resolution once at the sensor. This will enable the fine detail of a surface to be seen with greater detail. In the process of structure from motion this will help determine the features between images and result in a more detailed reconstruction.

Using the macro lens with the Canon DSLR, described in the previous section, we should be able to get more fine details of the surface. With more detail and features being seen on the object, we will be able to construct the mesh reconstruction much better because the feature matching in the SfM process should produce more matches and have better alignment. These lenses do not carry a high price, \$399, and are becoming very standard in every day photographic use in the world today. The \$399 price tag for this lens holds very reasonable to other lenses on the market that are not macro.

D. Graphical User Interface

A graphical user interface (also known as a GUI) is a way to take advantage of a computers graphical capabilities to make using a program easier to use. If done properly, the GUI should enable a user of the program to be free from learning complex languages, such as C++, Python, C# and so on. The developer of the GUI needs to know the programming language, but once developed, the GUI will eliminate that need.

In addition to the visual components, graphical user interfaces also make it easier to combine different elements of code into one setting. For instance, moving the CNC machine and taking photos with a camera can be done using the same GUI.

Some components of a GUI would be a pointer, icons, menus, and a desktop. The desktop will be the main display where all the interaction occurs while the pointer, icons, and menus will all be features inside the desktop and most likely will be selections for commands needed to run a program.

Below, in Figure 7, you will see the example GUI that was used for our system. The user can easily make any changes to the process by selecting key features without needing to understand programming languages. This will allow someone, like an artist, to easily operate the program and generate results needed.

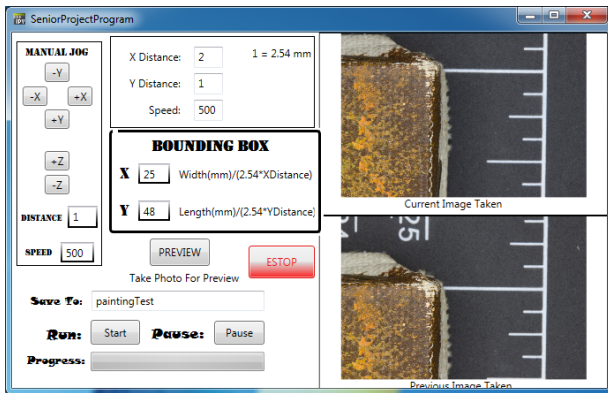


Fig 7. The Graphical User Interface shown here gives users the ability to manually jog the camera to different locations. The easy to use setup gives any user the ability to conduct a scan of an object while properly saving all images into a directory of choice.

III. PROCEDURE

Each component, discussed in the previous section alone cannot produce what is needed to reach the goal of an automated process for surface reconstruction. The task after choosing the components to use was then to integrate these parts and make them work together as a system.

A. Controlling CNC machine

Our first task was to be able to take the Probotix Fireball V90 CNC machine and find a way to control the movements in the X, Y and Z directions. There is a package available in Linux, to control CNC machines, called LinuxCNC.[19] This software proved difficult to use with our machine because it requires the use of parallel ports instead of the USB connection that the Probotix machines come with.

The Probotix Fireball V90 communicates through a controller that connects to the computer and the motors. The controller is connected to motor drivers which in turn drive the motors. The computer is connected to the USB connector. Fig 8 shows the schematic for such connections. In addition to the ability to control the motors, the controller box has the ability to install limit switches and other buttons for inputs and outputs.

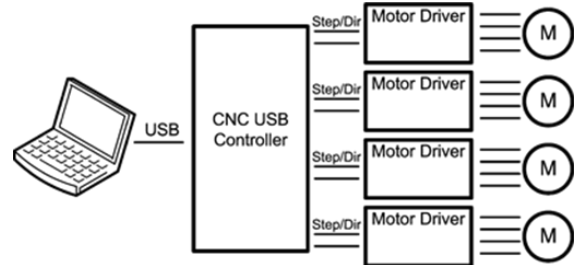


Fig 8. Schematic showing how the controller CNC machine is connected to the computer. The controller box connects the computer to the motor controls. It acts as the hub that all communication goes through for proper functionality.

Needing to find a software that can execute commands using the USB cable proved to be quite challenging. In fact, what was discovered was a company called Planet-CNC[20], which proved to be the only option to control this machine. Planet-CNC sells a license key to access their GUI and use their already generated program for under \$100.

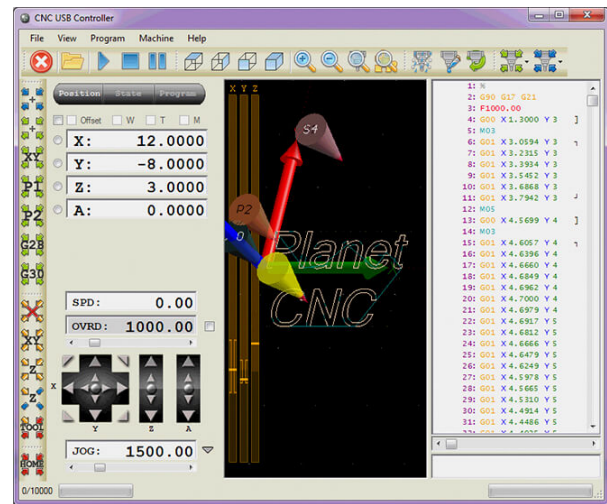


Fig 9. Planet CNC GUI that has the ability to control the Fireball V90.

Once we had the license key we were able to control the CNC machine using the Planet-CNC program. Unfortunately, this wasn't all that we needed. To be able to accomplish what was needed for our system we had to have total control of movements. Through some research online we were able to discover that some of the API for Planet-CNC was distributed online and available.[20] The code was written in the native Planet-CNC language of C# and was able to be opened with Microsoft Visual Studios.[20] It had basic movement commands and other safety features that we would want for our system.

B. IronPython with Visual Studio

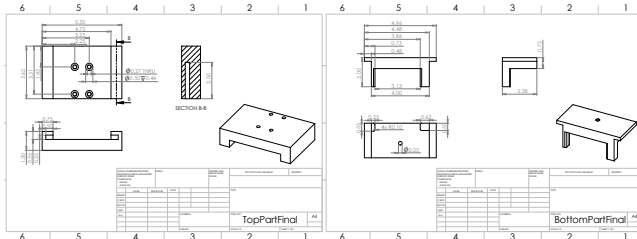
Since Python was the language of choice for this project, we needed to find a way to implement use of C# functions within Python environment. IronPython is an implementation of Python that enables the .Net framework to be understood while using Python on a window-based computer.[21] While using IronPython you can use scripts written in Python that interact with a .NET object like one may within C#.

After making use of IronPython, it was soon possible to call the C# API from Planet-CNC and use any of the commands as simple objects in a Python script. This permitted the use of any of the CNC developer commands as simple objects within a Python script.

C. Mounting the Canon Camera

With complete control over the CNC machine achieved, it was time to mount the camera onto the machine. The Fireball V90 was not designed to carry a camera, let alone a large DSLR camera that we are using. We needed to find a way to mount the camera and have it face in the downward direction toward the surface of the CNC machine.

The Fireball V90 has a small working distance between the tool mount and the table surface. In order to create the largest possible away from the table in the Z direction, it was crucial to create two brackets to keep the camera as high as possible. To do this a model was designed, and that model was then fabricated in a precision shop. The following is the CAD design for the brackets used to mount the camera:



Drawing 1: Design for brackets to mount camera to CNC machine.

As you can see in Figure 10, the first bracket attaches to the CNC machine and allows for the second bracket to slide downward into place.

The importance of the second bracket goes beyond having the most height from the tabletop. Having the second bracket will allow other camera models to be used in our proposed system. This opens the door for other imaging applications beyond its proposed use of the extraction of surface topography.

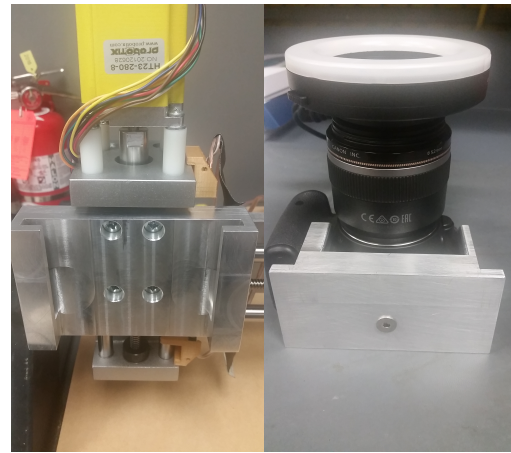


Fig 10. The Left image shows the first mount that has tracks on the sides to allow the second bracket, on the Right, to slide down into place. This will achieve the highest distance from the surface as possible, while allowing for multiple cameras to have their own bracket.

Once the brackets were fabricated, we could then test the mounting process for a good, tight fit and be able to see if the camera is indeed level and facing straight downward. We want the camera to be level and facing perpendicular to the surface to make sure that all images are being taken from a vantage point of directly overhead. In Figure 11, the final result is shown and shows the camera facing the table top surface.

To check the level of the camera, a small bullet-level was used to ensure the camera was in the correct place. It is important that the camera is level because in Structure-from-motion algorithms there are several areas that may become difficult to compute if the camera isn't staying level. The camera pose estimation process may become difficult if the images are being taken at an angle to the surface. Also, there may be shadows created if the camera isn't capturing from a straight downward view. This may cause errors in the reconstruction process by having features unrecognizable in the photos. These errors, though may be little, could cause the reconstruction to be worse. It may be beneficial to study the errors of the camera alignment, in order to determine the amount of error, if any, that may enter the process. For our study, the camera was properly aligned and level.



Fig 11. Image of camera successfully mounted onto the Fireball V90 with the macro lens and illuminator ring attached. The camera now will have the ability to move freely in the X,Y and Z directions while

D. Installing Limit Switches

Whenever a machine with limited movements is being operated, it is important to be able to stop the operation if an error occurs and it attempts to move beyond its physical limits. To eliminate any chance that the CNC machine moves beyond the limits, we obtained limit switches and mounted them so that whenever the machine moved to a certain point the limit switch would be enabled and the machine would stop all commands.

These limit switches needed to be soldered and electronically wired to the controller discussed earlier in the paper. Once they were working mounting brackets were 3D printed to hold them in their proper position. (Figure 12)

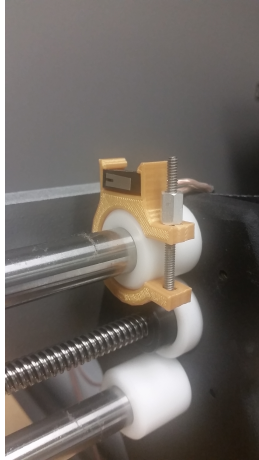


Fig 12. Limit Switch and 3D printed bracket to stop CNC machine if it runs off axis limit.

E. Automated Image Capture

The last controllable feature we need for this system is the ability to capture images at different increments, when commanded. In order to tackle this problem gphoto2 was used as a control for the camera.[22]

Gphoto2 is a command-line code for the package libgphoto2. This will allow the ability to use terminal commands or scripts to operate any camera functions just like you were holding the camera in your hands and changing the camera settings. We successfully pulled this into our Python environment by creating a windows-based script. We now could execute this script file whenever an image was needed to be captured in the control program.

F. Microsoft Visual Studio/GUI creation

With all of the software components successfully synthesized, the final step lay in integrating all of them together into a simple user interface. This was accomplished using Microsoft Visual Studio's WPF application, which is an easy way to build a GUI.

There were several key components needed in the GUI that will allow for the proper data to be collected. Figure 13 shows the final GUI that was generated using Microsoft Visual Studio.

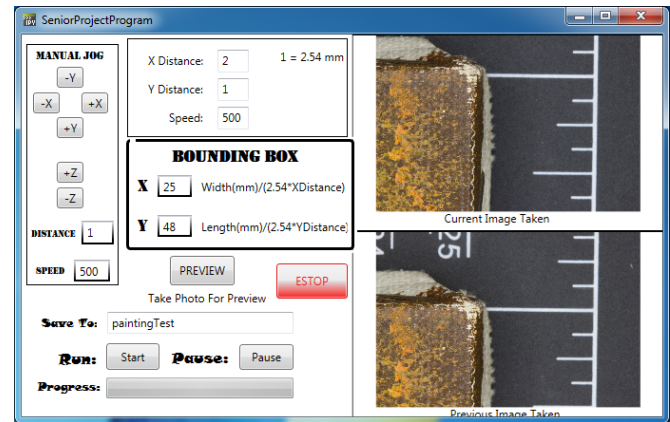


Fig 13. Final GUI created using Visual Studios. Components include manual movement, different X & Y distance movements, a bounding box scan and Photo previews.

Our main goal was to create a system that takes photos at given steps in the X and Y direction over the entire surface of the object. The first section that was created was the manual jog section. Buttons were created that, when pressed, would move the machine in the direction indicated. In order to move the machine a speed and distance per movement was needed, so two text boxes were created that the user could add those parameters. Manual movement would be used before the process would start to ensure proper alignment and starting point.

Manual jog alone does not satisfy the automation requirement. To accomplish that, a couple of features were added to complete a scan of the surface while taking an image at each interval.

When looking into the camera viewfinder, you may notice the screen is not square. The length in X is usually longer than the length in Y. In order to keep an 80-85% overlap in both directions you may need to alter the movement distance in the X and Y direction separately. To accomplish this, boxes for X and Y distance was created so movements would be different for each, if needed. The Bounding Box section will allow the user to input the correct amount of movements needed in the X and Y directions to scan the entire surface area. The formula for the calculation is written for the user to see, so as to limit the possibility of error.

As you can see in Figure 13, you can label the directory that you want your captured images to save to, as well as buttons to click to start and stop the scan.

The last component to talk about is the preview button. When starting the program, it is important to know the images being taken are properly exposed and in focus. One does not want to wait until all images are taken to see they are all inadequate for the task. In addition to the preview button, which is used before the program starts, a current and previous image preview will be displayed throughout the program.

When the Run button is pressed the CNC machine will move and the camera will take images at each location of movement. Those photos will then be stored in a directory for later use with the 3D reconstruction program of choice. Figure 14 shows the system together and in the beginning position, ready to scan. The illuminating ring attached to the camera is important because we want every photo to have the same

exposure and eliminate any shadows that may occur with uneven illumination.

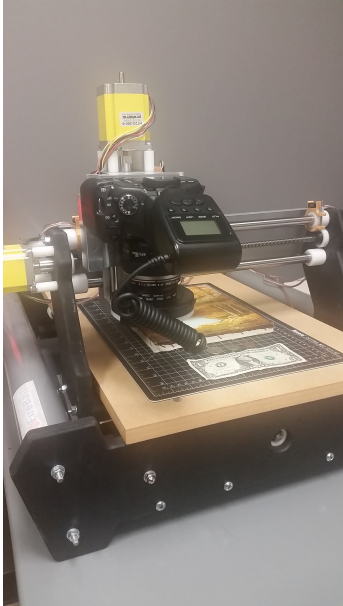


Fig 14. Complete system, ready to scan. Camera is facing downward at the beginning corner of the painting and will continue moving until all photos are taken. A typical painting 12 X 8 in will produce about 1300 photos with the field of view around an inch.

IV. ANALYSIS OF SELECTED RESULTS

Once the scan has completed, you now have most of the data needed for SfM reconstruction. The entire directory of images can be processed in any SfM program on the market. For the results given in this paper, we used Pix4D as the program to conduct the reconstruction. After processing, you will have a mesh reconstruction of the area scanned. In Figure 15, you can see the comparison between a photo and the reconstructed mesh. With a visual comparison between the two images it would be indicated that the process was highly accurate.



Fig 15. The top image shows a photograph of a dollar bills and 3 coins from the Canon t2i. In the bottom image you can see the reconstructed mesh that was processed through the images made by the system in Pix4D. The mesh is conducted in greyscale, although color could be looked at, this paper focused on surface measurements instead of color measurements.

Despite this, the reconstruction is imperfect, and in order to fully understand how accurate our proposed system is, further analysis needs to be conducted. Three separate accuracy models are needed to be measured to characterize the device performance. These will be discussed in the following.

A. Motor movement accuracy

Every CNC machine has claims that its movements are highly accurate. In a perfect world, every movement you make on a CNC machine will be true. If you tell the tool to move 1 mm, it would move exactly 1 mm. Of course we know this not to be the case. Most CNC machines are close, but it is important to be able to understand how close the movements are to perfect so we can ensure proper overlap is being conducted in the scan. We would like for the accuracy of the movements to be under 100 microns, for the purpose of our system. There is no need to be much more accurate than that, because structure-from-motion does not require the images to be captured at exact intervals. If we planned to capture the images with little to no overlap, then this accuracy would need to be much lower than the 100 microns goal we were hoping for. In order to test the accuracy of the mechanical movements in the x and y direction we had to conduct a couple different tests for reasons discussed later.

To get an accuracy level on the motor movements a penny was placed onto the surface of the CNC machine. We took a photo of the penny and then moved out 80 steps in the positive

x direction. We then returned back to the starting location by making 80 movements in the $-x$ direction and took another photo of the penny.

These two images were then put through a registration process using SIFT and ransac methods to produce a matrix that gives the pixel position difference between the two images in the x and y direction.

The same test was then carried out in the y direction to model the accuracy of the stepper motors in that direction as well. Table I shows that there was a much larger difference in the y direction than in the x direction.

Table I: Non-Fixed Measurements of Stepper Motor Accuracy

Non-Fixed X Movement			
Direction	Avg Pixels	Avg μm	Std
X	1.86	12.64	1.53
Y	-13.15	-92.82	3.61
Non-Fixed Y Movement			
Direction	Avg Pixels	Avg μm	Std
X	-0.10	-0.64	1.03
Y	12.61	89.58	3.37

The large inaccuracy in the y direction for both movements led to further analysis. What was discovered was the controller box, and motors created a lot of vibrations on the CNC table surface. The thought was the error may be coming from these vibrations, as the penny may be moving during the experiment.

In order to tell whether the accuracy error was coming from the motor movements or the vibrations, a second test was conducted. In this test, a target was drawn onto the surface of the CNC machine. The same movements were made and the results were again processed in the same accordance as the prior test.



Fig 18. Image on the left is the photo of the target before movement of the motor and the image on the right is the photo of the target after the motor movements. The stationary targets reduced the amount of error received by eliminating the vibrational error that was happening.

Using the two images received in this experiment, as seen in Figure 18, the same analysis was conducted and the following results were obtained.

Table II: Fixed Measurements of Stepper Motor Accuracy

Fixed X Movement			
Direction	Avg Pixels	Avg μm	Std
X	-1.37	-9.74	0.83
Y	-7.52	-53.43	4.39
Fixed Y Movement			
Direction	Avg Pixels	Avg μm	Std
X	-0.37	-2.62	0.92
Y	9.66	68.62	4.93

Table II shows that the average difference for the y did reduce by about 40 percent, but the values for x stayed near the same for both experiments. The problem was pinpointed to the structure of the CNC machine. The two arms of the machine are made of manufactured wood and is attached to the sliding rails by more manufactured wood. When we put any pressure in the y direction, onto the camera, the entire mount would shake. This led to the conclusion that the error in the y direction was attributed to the structure of the machine itself. The resolution to this problem is discussed later in the future work section.

Overall, the CNC machine is fairly accurate. While it is not a perfect machine, it is good enough for the way it is used for these purposes. There is no need to be more accurate in movements than this CNC machine makes, because the Structure from Motion process does not require perfect movements when taking the images. We conduct at least 80-85% overlap in our images, so being off by 50-60 microns at this capture height will be insignificant in the process.

B. Assessment in accuracy of reconstructed mesh

Once we have a reconstructed model the real significant data we want to know is how well the reconstructed model matches reality. In other words, how close is the model to real life measurements. If an inch is measured on the 3D model, is that measurement actually an inch.

The three different accuracy models we want to measure are the x, y and z axis of our 3D reconstructed mesh. It is assumed that aberrations in the optics of our device are symmetrical across the x and y plane, this means that they are considered linear and will have the same % error. This then means we can measure the x and y axis together and then assess the z axis separately.

1. Analysis of X and Y accuracy

In order to attain the accuracy of the reconstruction, across the imaging plane in the x and y direction, a test was conducted between a known measurement in real life and that same corresponding distance in the reconstructed mesh. The results of the dollar bill scene, in Figure 20, you will see that there is a grid behind the currency. Each of the grid squares is known to be $\frac{1}{2}$ inch on a side. To make things simplistic we took two squares and labeled our truth value at 1 inch.

The measurements were marked by visual inspection, and therefore, the error of this measurement should be added into the final error percentage. Multiple measurements were made in each trial to, hopefully, achieve an average measurement that lowers the error induced from human measurement.

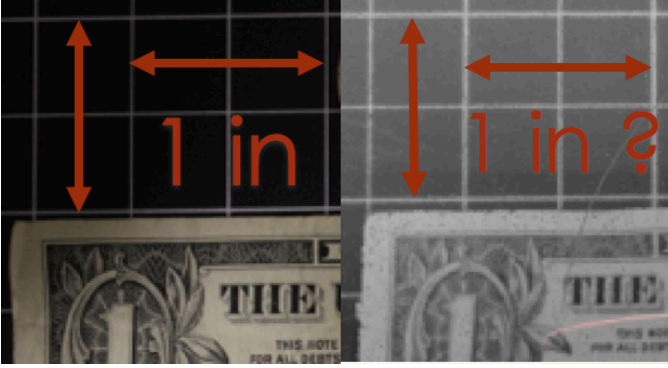


Fig 20. Image on the left models the ground truth value in distance between 2 grid squares and the right models the corresponding value in x and y distance.

We then used a conversion factor to scale between inches and pixels (making the assumption that the SfM reconstruction retains approximate linearity across the imaging field). With this conversion factor we then took 5 measurements between two grid boxes in different locations throughout the 3D mesh. This was conducted 5 separate times and the results are seen in Table III.

Table III: Error in X and Y for Reconstruction Compared to Truth

Error in X,Y			
Trial	Avg Trial Measurement	Converted Measurement (in)	% Error
1	29.0182	0.9315	-6.85%
2	28.9635	0.9297	-7.03%
3	29.1317	0.9351	-6.49%
4	29.0411	0.9322	-6.78%
5	29.1562	0.9359	-6.41%
Conversion Factor x,y:		0.0321	in/Vx
Truth Value:		1	Inch
Avg % Error:		-6.71%	

As you can see the result was an average of 6.71% or .06 inches smaller in our 3D mesh than what a true measurement would be. Further analysis could be conducted to determine the exact reason for this error. If we know this error to be true, then we could scale the entire 3D mesh by our scale factor and this will give us a very close representation of the x and y size of the 3D reconstructed model.

2. Analysis of depth accuracy (Z axis)

For our project, the depth accuracy on the reconstruction is the most significant to analyze. It also was the most challenging. As discussed earlier in the paper, there are many factors that are involved when going from taking images using our system and the final 3D reconstructed model from Pix4D. One of the greatest challenges facing the SfM reconstruction is the registration process for the features being detected. Being off by even a pixel or two in the registering process can result

in a major error in depth.[3] This can be shown by the following:

$$\Delta z = |d_{z2} - d_{z1}| = |(\Delta h_1 / 2 \tan \theta_1) - (\Delta h_2 / 2 \tan \theta_2)|$$

where Δz is the error in depth, d_z is the calculated depth, and $\Delta h_1, \Delta h_2$ are the horizontal differences between two registered features in images. The error in depth can be directly related to the Δh calculated in the registering process. If Δh is smaller than truth, then the calculated depth will be smaller than it should be. Likewise, depth will be larger if the registration gives a larger Δh than it should.

This can be seen in the diagram below in Figure 22. This is one reason that the error in the z axis is expected to be worse than the x or y axis.

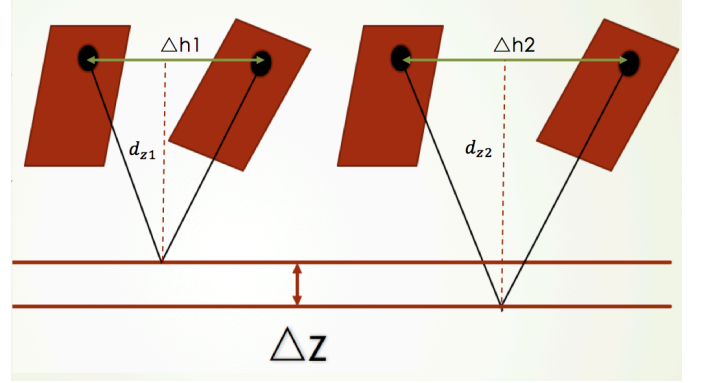


Fig 22. This shows the depth change that could happen by even the slightest registration error. A pixel or two in the registration process can alter the depth of the 3D reconstruction significantly.

The left side shows the convergence between two registered images and the right shows what will happen to d_z if Δh is increased. Thus Δz will be the error that is created from this shift in the registration process.

In order to calculate the depth error of our process we had to have a truth value of a surface known to us to be accurate. This proved quite challenging to find something to give us an accurate measurement from one of our surfaces.

In the end, a profilometer was used to get topographical data on the \$0.25 piece seen in Figure 15 from earlier in the paper. This coin was measured using the profilometer and the results were compared to our 3D reconstruction results.

The profilometer used to compare the result with was the PS50 optical profilometer made by Nanovea. The PS50 is the most advanced compact profilometer available and is capable of 1.5-micron resolution in the XY planes.[23]

The profilometer gave an xyz file that had all the point cloud information it measured. An xyz file is primarily associated with coordinates and can be seen as points in space. We compared the xyz file of the profilometer to the xyz file our model came up with.

The data, between the two, could not be directly compared to each other until proper registration was conducted. This was to make sure the two sets of data were in the same space. This process combined several different image processing techniques to ensure the data was being handled properly. The end results after proper registration revealed that our model

was consistent with the results from the profilometer. The results can be seen in the following figure:

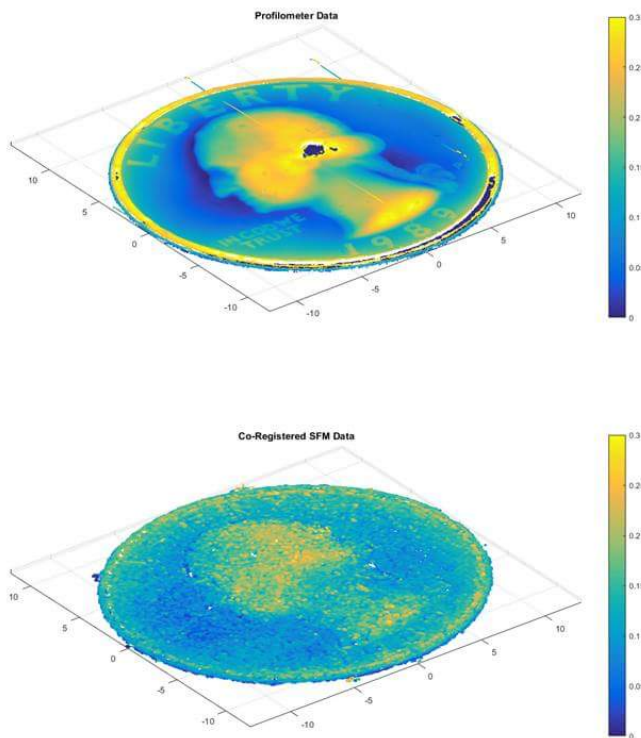


Fig 23. Surface measurements of a profilometer, top, and the proposed system using SfM algorithm, bottom.

This gives us a visual representation of how close the two point clouds are to each other, but without knowing actual values it would be unwise to give an assessment on the accuracy.

To quantify the depth accuracy between these data sets, further analysis was needed. Taking a slice of each point cloud would enable us to plot the two sets of data at those regions on the same graph. In Figure 24, you can see that the slice was taken from the top half of the quarter, in each case, and plotted together in the graph on the right. We can tell that the profilometer data has less noise than the system created here does, however, the data shows that our system actually generated a point cloud that holds close to a system that costs over \$50,000. Table IV shows that the accuracy is within an average of 57 microns RMSE.

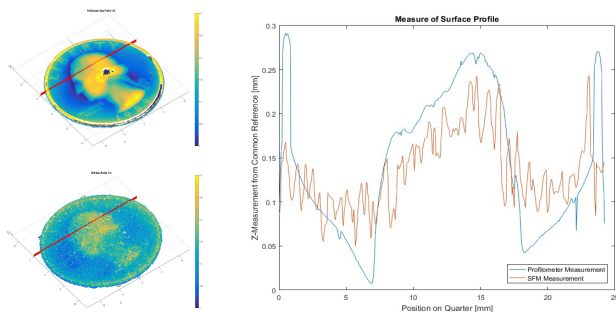


Fig 24. Results showing the accuracy between a high-end profilometer and our low-cost system. A slice of the quarter is taken from each and plotted together. A slight shift of the data may result in a more accurate assessment.

Table IV: Error in Z (Depth)

Error in Z (Depth)		
RMSE	57.1	μm
Percent Difference	18.7	%

The same process was conducted at another slice in the quarter. This was done to ensure the results hold true at different points along the scene. Figure 26 and 27 will show the results at a location along the lower part of the coin. They show that the results stay close to the same at this region as the prior one was.

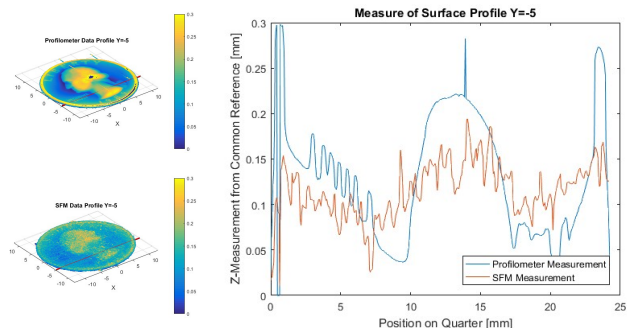


Fig 26. This shows a slice being taken from a lower section of the quarter, for each data set, and plotted together to see the similarity between the two 3D reconstruction.

Table V: Error in Z (Depth)

Error in Z (Depth)		
RMSE	61.1	μm
Percent Difference	19.3	%

The results that were received from our system resulted in an average depth difference of between 18 and 19%. While this is not perfect, it can give us enough data to be able to come up with a scaling factor when using this system. For instance, in both measurements in Figures 24 and 25, the data suggests there is a reduction in the modulation depth. Scaling may be difficult if the reduction in modulation depth does not hold throughout the entire surface. There may need to be more research to determine the proper way to scale the models in the z plane to match the truth data.

We know that the error in depth is most likely caused from problems in the SfM algorithm and registration problems that occur. There is further research ongoing involving SfM and the ability to limit the error in depth across a scene being reconstructed.[3] As these algorithms increase in accuracy so too will this system.

C. Advantages

Our system carries many advantages compared to other systems in the same field. Our design is relatively small and portable and can be easily moved or transported to different locations. This system also is relatively inexpensive and can be afforded by almost anyone. We offer a non-intrusive way to

capture the surface, which may be useful for anyone working with delicate items. Lastly, we offer a design that is non-complex and a GUI that is very easy to use.

D. Challenges

As with any product on the market there is always challenges that may occur and need to be overcome. Our system carries a few of these. The CNC machine used only carries a surface for imaging of 18x24 inches. Another challenge to be aware of is the vibrations which are propagated throughout the CNC machine by the nature of its build materials and the motors which are used. These are not ideal during the task of precision macro imaging, and some sort of dampening would be ideal in order to increase the accuracy of this system. Another challenge to be aware of is the vibrations and structure of the CNC machine when conducting imaging.

V. FUTURE WORK

There are many areas of research that could continue further in regards to this proposed system:

1. Studies could be done on the accuracy of the motor movements on the CNC machine. A study could be done to see if structural support would limit the error received in the x and y movements.
2. Another study that may need further analysis is a study on the SfM algorithm and if the depth error is coming from the algorithm or from the images being output from our device. This would involve knowing all the components and processes that are being done “under the hood” of the algorithm itself. If we can find the source of the error, we may be able to lower it and achieve better results.
3. Future work could be done on the GUI created. There could be more functions and better handling of the controls. This would involve creating an API from scratch that would control the motors and other functions of the CNC machine.
4. Further work could be done to study the color differences in the 3D reconstruction compared to the actual color of the object or scene being imaged.

VI. CONCLUSION

Our goal at the end of this experiment was to design and implement an imaging system that would be low-cost and effective in reconstructing a 3D model of an object. Making use of a CNC machine, that was available to us, we were able to make use of an API in order to control movements. We also designed a bracket to attach a camera to the CNC machine, in order to capture images using the technique of Structure from Motion. Our final achievement was creating an easy to use GUI to allow anyone to properly control the system. In the

end, we were able to achieve this goal by creating a system that is under \$3,000 and is comparable to other high end devices such as a profilometer. Our accuracy measurements were within 50-60 microns of a high-end measurement device, such as a profilometer.

VII. REFERENCES

- [1] Westoby, M. J., Brasington, J., Glasser, N. F., Hambrey, M. J., & Reynolds, J. M. (2012). ‘Structure-from-Motion’ photogrammetry: A low-cost, effective tool for geoscience applications. *Geomorphology*, 179, 300-314.
- [2] Fonstad, Mark A., et al. "Topographic structure from motion: a new development in photogrammetric measurement." *Earth Surface Processes and Landforms* 38.4 (2013): 421-430.
- [3] Nilosek, David R.; Walvoord, Derek J.; Salvaggio, Carl, Assessing geoaccuracy of structure from motion point clouds from long-range image collections, *Optical Engineering*, 53, 11, pp. 113112-1-113112-10 (November 2014).
- [4] Sayles, R. S., & Thomas, T. R. (1978). Surface topography as a nonstationary random process.
- [5] Valentino, J., & Goldenberg, J. (2003). Introduction to computer numerical control (CNC). Englewood Cliffs: Prentice Hall.
- [6] Perone, C. S. (2009). Pyevolve: a Python open-source framework for genetic algorithms. *ACM SIGEVOlution*, 4(1), 12-20.
- [7] Koenderink, J. J., & Van Doorn, A. J. (1991). Affine structure from motion. *JOSA A*, 8(2), 377-385.
- [8] Faugeras, O. D., & Lustman, F. (1988). Motion and structure from motion in a piecewise planar environment. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(03), 485-508.
- [9] Stout, K. J., & Blunt, L. (2000). Three dimensional surface topography. Elsevier.
- [10] <http://www.thephotoexpress.com/why-start-macro-photography/>
- [11] <http://www.its-inc.com/page/3d-surface-topography-30.html>
- [12] "Computerized Numerical Control". www.sheltonstate.edu. Shelton State Community College. Retrieved March 24, 2015.
- [13] https://en.wikipedia.org/wiki/Numerical_control
- [14] <http://www.mfg.mtu.edu/cyberman/quality/sfinish/terminology.html>
- [15] <http://www.brnt.eu/phd/node16.html>
- [16] <https://cloud.pix4d.com/store/>
- [17] <http://www.probotix.com/V90-MK2>
- [18] https://en.wikipedia.org/wiki/Digital_single-lens_reflex_camera
- [19] <http://www.linuxcnc.org>
- [20] <http://planet-cnc.com>
- [21] <http://ironpython.net>
- [22] <http://gphoto.sourceforge.net>
- [23] nanovea.com/wp-content/themes/wp-nanovea/brochures/profilers.pdf