

University of Pennsylvania
Center for Sensor Technologies
Philadelphia, PA 19104

SUNFEST REU Program

Archeoviz: Improving the Camera Calibration Process

NSF Summer Undergraduate Fellowship in Sensor Technologies
Jonathan Goulet (Computer Science and Engineering) – University of Pennsylvania
Advisor: Dr. Kostas Daniilidis

ABSTRACT

The reconstruction of archaeological scenes as three-dimensional graphical models can provide a very powerful tool for the archaeologists studying them. The purpose of the Archeoviz project is to develop a program that archaeologists can use in the field to create these three-dimensional models themselves, accurately and easily, using two-dimensional photos of the scenes. This paper describes my development of a program to make the process of camera calibration, necessary before any model building, as fast and easy as possible for the user. This calibration process requires locating and recording image coordinates of numerous small markers in the scenes. Originally, this locating had been quite time-consuming and had to be done separately from any calibration calculations. I have developed an independent C++ program that requires users to locate only six of these markers themselves. The program will then automatically locate the remaining markers for the user and calculate the calibration data. This program includes a user-friendly zooming interface and a file browsing system for retrieving and loading the images and for saving the calibration data obtained. In addition, an extension for the program will allow users to estimate the location of the markers in the second image of the stereoscopic pair from the calibration data for the first. Also, in order to obtain more accurate three-dimensional models, the reconstruction will incorporate stereo pairs from multiple views of a scene. Through another extension, the calibration data from the first of these pairs can be used to predict the location of all markers in all subsequent stereo pairs. Through these various improvements, my program greatly speeds up this calibration process and allows archaeologists to quickly move on to constructing these valuable models.

Table of Contents

1. INTRODUCTION.....	95
2. MATHEMATICAL COMPONENTS.....	95
2.1 Camera Calibration.....	95
2.2 Stereoscopic Reconstruction.....	97
3. THE ORIGINAL ARCHEOVIZ IMPLEMENTATION.....	98
4. METHODS OF IMPROVING THE CALIBRATION PROCESS.....	99
4.1 The Six Points Algorithm.....	99
4.2 Predicting the Right Projection Matrix from the Left.....	99
4.3 Predictions Using the Location of the Tripod.....	103
5. RESULTS AND CONCLUSIONS.....	106
6. ACKNOWLEDGMENTS.....	106
7. RESOURCES.....	106
APPENDIX A: THE SOFTWARE IMPLEMENTATION.....	107

1. INTRODUCTION

The study of archaeological sites can give us both a link to the world of the past and a better understanding of that of our present. The reconstruction of these sites as three-dimensional graphical models can bring them directly to anyone with access to a computer.

The goal of the Archeoviz project is to provide full three-dimensional reconstructions of the pre-Columbian site of Tiwanaku, located in Bolivia, for the Museum of the University of Pennsylvania. The project also works to develop a program for archaeologists to use in the field that will allow them to create these models themselves, quickly and accurately, from two-dimensional photographs of the scenes they are studying. This program is intended to be as user-friendly and as easy to use as possible so that any archaeologist working in the field can, without any prior experience, create these models. These models would then provide them with the ability to study the sites and to collect valuable data from the scenes that they would not have been able to gain from the original two-dimensional photographs. In addition, the archaeologists would be able to create these models in just a few minutes, as opposed to the days or even weeks it would take them to create the accurate topographical maps that they now use for similar data taking purposes.

The first step in this reconstruction is the process known as camera calibration. This report describes my development of an independent program that will make this calibration as easy and as fast as possible for the archaeologists so that they can quickly proceed to creating these valuable models. The methods used to improve the calibration process are discussed in the main body of the report and a detailed description of the software implementation is provided in Appendix A.

2. MATHEMATICAL COMPONENTS

The two major mathematical components of the reconstruction process are camera calibration and stereoscopic reconstruction. My portion of the project focused mostly on the former, but I will give a short explanation of the latter as well.

2.1 Camera Calibration

In order to construct a three-dimensional model using only two-dimensional photographs, it is necessary to find a transformation from the two-dimensional image coordinates of points and features in the picture to some three-dimensional coordinate system in which these features can be represented. This process, known as camera calibration, is done in two steps, transforming from the two-dimensional image coordinates to the three-dimensional coordinates relative to the frame of the camera and from those to the three-dimensional coordinates relative to some fixed world system. The parameters describing the first transformation are known as the intrinsic parameters of the camera and those describing the second are called the extrinsic parameters. Figure 1 shows the three coordinate systems of interest in our calibration problem.

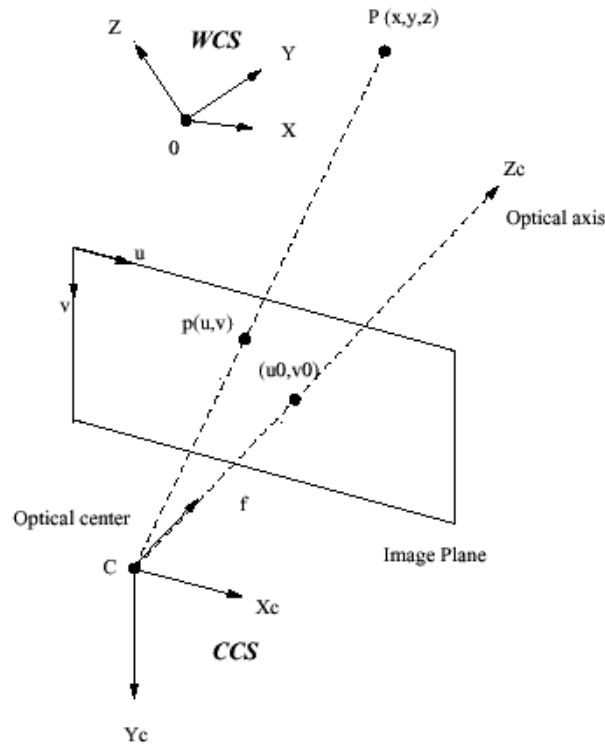


Figure 1: The world, camera, and image coordinate systems.
 Figure taken from www.csd.ucl.ac.uk/~cgarcia/JANUS/pinhole.gif [1]

Four intrinsic parameters must be recovered in the calibration of the cameras. The first two describe the distance from the origin of the camera coordinate system, the optical center of the camera, to the image plane. These two parameters, denoted by s_u and s_v , also take into account the scale factors, pixels per unit length, in both the u and v directions in the image plane. The second pair of parameters describes the point in the image plane where the optical axis, the z -axis in the camera's coordinate system, intersects the image plane. These are denoted by u_0 and v_0 and are measured relative to the origin of the image coordinate system, in the upper left corner of the image plane. If these parameters are placed in a matrix, the transformation from image coordinates to camera coordinates can be described with the following equation, where the vector of homogeneous image coordinates is denoted by $(\mathbf{u}, \mathbf{v}, \mathbf{w})$ and the vector of homogeneous camera coordinates by $(\mathbf{x}_c, \mathbf{y}_c, \mathbf{z}_c, \mathbf{w}_c)$ and where \sim denotes equality up to a scale factor:

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} \sim \begin{pmatrix} s_u & 0 & u_0 & 0 \\ 0 & s_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \\ W_c \end{pmatrix}.$$

This equation gives the location in the image plane onto which a point with coordinates

(x_c, y_c, z_c, w_c) relative to the camera's coordinate system will be projected.

The extrinsic parameters of the camera are a rotation and a translation that relate the three-dimensional coordinate system centered at the camera to some fixed three-dimensional world coordinate system. Let the 3x3 matrix of rotation be denoted by \mathbf{R} and the 3x1 matrix of translation by \mathbf{t} . The transformation from camera to world coordinates can then be described by the following equation, where the camera coordinates of a point are denoted by (x_c, y_c, z_c) and the world coordinates of the point by (x_w, y_w, z_w) :

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} = \mathbf{R} \begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix} + \mathbf{t}.$$

This transformation describes the representation, in the camera coordinate system, of the point with coordinates (x_w, y_w, z_w) in the world coordinate system. Equivalently, using homogeneous coordinates to represent a point in the two coordinate systems, the equation becomes:

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \\ W_c \end{pmatrix} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ W_w \end{pmatrix}.$$

By combining the above two transformations, we can obtain a transformation from the homogeneous world coordinates of a point to the homogeneous image coordinates of the projection of that point onto the image:

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} \sim \begin{pmatrix} su & 0 & u0 \\ 0 & sv & v0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ W_w \end{pmatrix} \rightarrow \begin{pmatrix} u \\ v \\ w \end{pmatrix} \sim \mathbf{M} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ W_w \end{pmatrix}.$$

In this equation, \mathbf{M} is a 3x4 matrix that will be referred to as the projection matrix. My program's method for camera calibration recovers this matrix and extracts from it all of the intrinsic and extrinsic parameters. This matrix will be necessary for the stereoscopic reconstruction step of the process.

2.2 Stereoscopic Reconstruction

The method used in the Archeoviz project to create the three-dimensional models is known as stereoscopic reconstruction. This process uses multiple, slightly different, two-dimensional images of the same scene, for which camera calibration has already

been completed, and exploits these differences to create a three-dimensional model of the scene. This is the method that the human visual system employs in using the two slightly different pictures on the retinas to create a three-dimensional view of the world. The basic manner in which this process operates uses the projection matrix obtained in the camera calibration step. With this matrix, a ray in the three-dimensional world coordinate system can be associated with each point in the image plane. By intersecting the rays obtained from various calibrated images of the same scene, we can obtain the three-dimensional location of image points and features, and a three-dimensional model can be constructed.

3. THE ORIGINAL ARCHEOVIZ IMPLEMENTATION

In the Archeoviz project, the calibration of the cameras is accomplished through the use of markers that are placed in each of the scenes to be reconstructed at locations with known coordinates relative to the fixed world system. The world coordinates of these locations are measured using the Theodolite laser measurement system. This system uses a laser to accurately judge the distance to any point and, using that data, gives the x , y , and z coordinates of the point, corresponding to the distance east and north and the elevation from a set origin, respectively. The input to the calibration process is a text file containing these x , y , and z coordinates, along with the image to be calibrated. Calibrating requires that the markers in the image be located and their two-dimensional image coordinates recorded. Then, from the correspondences between the coordinates, we solve a system of linear equations, using linear least-squares techniques, to recover the projection matrix.

A program to accomplish the reconstruction of the scenes was created in the summer of 2002 as part of the Archeoviz project. However, the calibration process in this original implementation was quite time-consuming and difficult. In this original implementation, the user had to locate each of the 20 or 25 markers in the picture one by one, using a program such as Adobe Photoshop, and input their image coordinates into a text file manually, in a similar format to the text file containing the world coordinates of the points. The user had to make sure that these points were entered in a very specific order, corresponding to the order in which the markers were represented in the text file containing the world coordinates. This order was determined by the numbering assigned to the markers in hand-drawn diagrams that the archaeologists had created of the scenes they wished to be reconstructed. Since these markers are often quite small and are sometimes quite difficult to find, recording the image coordinates could take quite a long time. Once the user had found all of the markers in a particular scene, the text files containing the coordinates had to be taken to a program that would compute the projection matrix. Finally, the user would take a text file containing this projection matrix, along with the other text files created, to the reconstruction program, which would create the three-dimensional models. My major goal during my time on the Archeoviz project was to create a program to make this calibration much faster and easier for the user.

4. METHODS OF IMPROVING THE CALIBRATION PROCESS

4.1 The Six Points Algorithm

In the original implementation of the program, the most time-consuming part of the calibration procedure was locating the markers. Thus, my major focus was on reducing the number of markers that users had to find themselves, and on enabling the program to automatically find as many of these markers as possible. The projection matrix that needs to be recovered has 12 entries, and each correspondence between world and image coordinates provides two equations on these 12 elements. Consequently, six is the minimum number of correspondences that need to be known, and thus the minimum number of markers that need to be located, in order to recover a projection matrix for the scene. Therefore, my first major step in improving the calibration process was to implement an algorithm that requires users to find just six of these markers, then calculates an estimate of the projection matrix and, using that estimate, predicts the image coordinates of the remaining markers from their known world coordinates. This method reduces the number of markers that users must discover themselves from 20-25 in each picture to six.

My first step was to prototype this algorithm in Matlab, and to test how well the estimates of the markers worked with different combinations of the six markers first selected. In the majority of cases, the estimates for the remaining points were quite accurate if the six markers selected were relatively sparsely spread throughout the picture. However, certain degenerate point configurations caused the method of prediction to become quite inaccurate. This occurred mostly when the six points selected lay in a formation that was close to being planar in the three-dimensional world coordinate system. If these degenerate configurations were avoided, however, my approach would predict the majority of the remaining markers within about 30 pixels in both the u and v directions in the image coordinate system. Thus, I chose this value as the radius of the circle that I would implement my program to draw around the predicted locations of the markers to help the user find them. This value of 30 pixels is quite small in relation to the size of the entire image, and so these predictions are quite close, relatively, to the actual location of the markers.

As the above results illustrate, this new method provides an accurate way of predicting the location of the remaining markers in the scene from the known location of any six markers that do not lie in a planar formation. This will allow the program to automatically discover the location of all but the first six markers for users. Consequently, this method provides a great improvement over the original calibration implementation by significantly shortening the time needed to complete the calibration through reducing the number of markers that users must locate themselves.

4.2 Predicting the Right Projection Matrix from the Left

As mentioned earlier, the method used in the reconstruction program to produce the three-dimensional models is stereoscopic reconstruction. Thus, for each scene to be

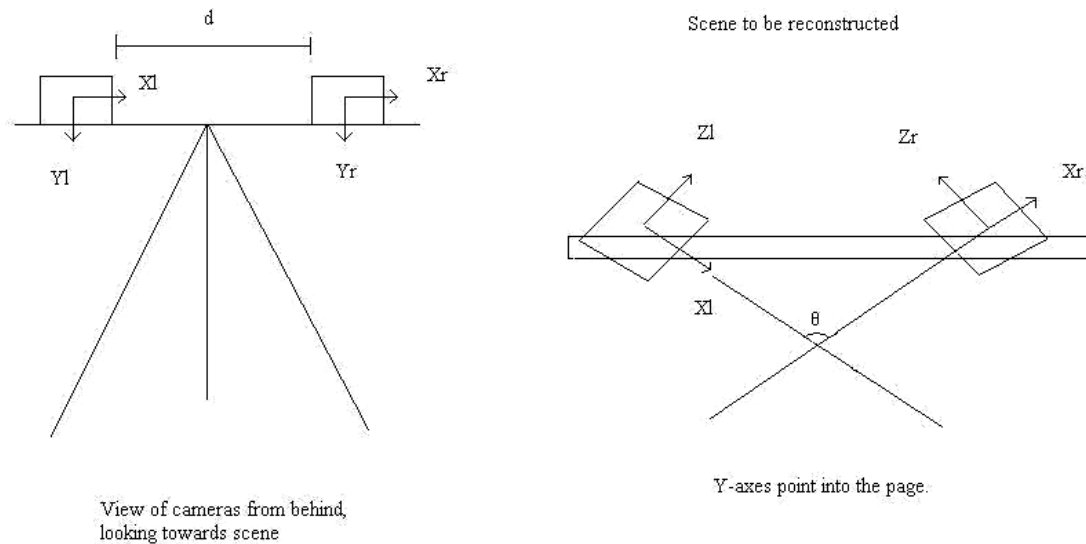
reconstructed, it is necessary to complete calibration and to recover the projection matrix for each of two images, called the “left” and “right” images, corresponding to the relative locations of the cameras at the time the photos are taken. In my next improvement to the calibration process, I have implemented a new method that will use the projection matrix for the left image, obtained using the method described above, along with the information relating the relative location of the cameras, to estimate the projection matrix for the right image and thus predict the location of all markers in this second image. This will further speed up the calibration process, as it allows users to calibrate both images at once, as well as eliminating their need to find any markers in the right image by themselves. Figure 2 shows a sample pair of stereo images from one of the archaeological sites.





Figure 2: A stereo pair of images, the left above and the right below, illustrating the slight differences between the two views.

In this estimation, I have taken advantage of the information provided by the archaeologists who had taken the photos of the scenes in Tiwanaku regarding the setup of the cameras for the scenes. For all of the scenes to be reconstructed from Tiwanaku, the left and right cameras were set up on a common tripod, on opposite ends of a horizontal bar that sat atop the tripod. They were separated by a short distance, generally about one half meter, and they were moved to rotate only slightly to get a better view of the scene. This setup is shown in Figure 3, where the orientations of the camera coordinate systems relative to the physical cameras are apparent. In addition, this diagram illustrates that the only movements for the cameras correspond to a rotation about the y-axis of the camera coordinate system. From this diagram it is also apparent that the transformation between left and right camera systems can be modeled as a rigid transformation, with a translation along the x-axis of the camera system and a rotation about the y-axis of the same system.



The Z-axes, along the optical axes of the camera, point into the page, toward the scene.

Figure 3: Geometric representation of camera coordinate systems, showing the relationship between left and right cameras.

Thus, two parameters will fully describe this transformation: the translation distance, d , and the angle of rotation about the y -axis of the camera system, θ . Consequently, this transformation can be modeled, mathematically, by the following equation, in which the coordinate vector of a point relative to the left camera frame is represented by \mathbf{X}_l and that of a point relative to the right frame by \mathbf{X}_r :

$$\mathbf{X}_l = {}^l\mathbf{R}_r * \mathbf{X}_r + {}^l\mathbf{t}_r$$

where:

$${}^l\mathbf{R}_r = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix} \text{ and } {}^l\mathbf{t}_r = \begin{pmatrix} d \\ 0 \\ 0 \end{pmatrix}.$$

By adjusting these parameters appropriately, we can find the correct transformation from left camera system to right camera system. Combining this with the transformation from left camera system to world system, which is acquired by extracting the extrinsic parameters from the projection matrix of the left image, we can obtain an estimate of the transformation from world system to right system. If \mathbf{X}_w represents the coordinate vector of a point in world coordinates, this transformation relating world and left camera systems can be modeled, as before:

$$\mathbf{X}_l = {}^l\mathbf{R}_w * \mathbf{X}_w + {}^l\mathbf{t}_w.$$

Using this and the prior relation, we can obtain the following transformation relating the world system and right camera system:

$$\mathbf{X}_r = {}^l\mathbf{R}_r^{-1} * {}^l\mathbf{R}_w * \mathbf{X}_w + {}^l\mathbf{R}_r^{-1} * ({}^l\mathbf{t}_w - {}^l\mathbf{t}_r).$$

Thus, using this formula, we can acquire an estimate of the extrinsic parameters for the right image. In addition, we assume that the intrinsic parameters of both cameras are the same. This assumption is based on data taken from many of the images from the Tiwanaku site, as well as the knowledge provided by the archaeologist that the zoom, and thus the focal length, of the cameras are kept relatively constant throughout the process of photographing. Therefore, using the estimated right extrinsic parameters, and setting the right intrinsic parameters to be the same as those obtained from the left projection matrix, we can estimate the right projection matrix and use it to predict the location of all of the markers in the right image. This will provide another method of speeding up the calibration, as users will not be required to locate any of the markers in the right image themselves.

4.3 Predictions Using the Location of the Tripod

In order to get more accurate three-dimensional models of the scenes, and so that these models can be viewed from many different viewpoints, the archaeologists have taken stereo pairs of images from many points of view for a given scene. Therefore, to capture a full 360° view of a scene, multiple stereo pairs of images must be calibrated. In my next improvement to the calibration process, I have implemented a method of predicting the projection matrices for all stereo pairs of images of a given scene using the calibration information obtained from just the first stereo pair. This is achieved through the inclusion of a few more pieces of data to be taken by the archaeologists in the field. In addition to recording the location of the markers in the fixed world coordinate system, the archaeologists in the field will have to measure the world coordinates of three fixed points on the tripod for each camera location that they use for the scene. Using these tripod measurements for each point of view, we can construct a transformation between the tripod positions. This transformation, along with the projection matrices obtained through the above method for the first stereo pair, can then be used to obtain an estimate for the projection matrices for all subsequent stereo pairs of images of the scene.

Using the knowledge of the world location of at least three points on the same rigid body, in this case the tripod housing the cameras, we can construct a rotation and a translation that relate the two positions of the rigid body. If \mathbf{X}_i^1 represents the coordinate vector of one of the measured points on the tripod at the first location, and \mathbf{X}_i^2 the coordinate vector of the same point at the second location, both measured relative to the world coordinate system, and \mathbf{X}_{avg}^1 and \mathbf{X}_{avg}^2 represent the average values of all points at each location, then the \mathbf{R} and \mathbf{t} that satisfy the following equations will give the desired transformation.

$$\begin{aligned} (X_{i2} - X_{avg2}, \dots, X_{n2} - X_{avg2}) &= \mathbf{R} * (X_{i1} - X_{avg1}, \dots, X_{n1} - X_{avg1}) \\ \mathbf{t} &= -\mathbf{R} * \mathbf{X}_{avg}^1 + \mathbf{X}_{avg}^2 \end{aligned}$$

The above rotation and translation can be obtained using Singular Value Decomposition, among other methods. In addition, if it is assumed that the cameras are attached rigidly to the tripod and stay in the same position relative to the tripod when it is moved from location to location, this rotation and translation will also describe the movement of the camera body and thus of the camera coordinate system attached to it. If the calibration has been completed and the projection matrices are known for the first stereo pair of images, these matrices, along with the above transformation, can be used to get an estimate of the projection matrices for any pair of stereo images for which the location of these fixed points on the tripod is known. Using the \mathbf{R} and \mathbf{t} obtained above, this estimation can be accomplished mathematically by the following method, where \mathbf{M}_1 is a 3x3 matrix containing the first three columns of the known projection matrix, \mathbf{m}_1 is a 3x1 matrix containing the last column of this known projection matrix, \mathbf{M}_2 is a 3x3 matrix containing the first three columns of the projection matrix we wish to estimate, and \mathbf{m}_2 is a 3x1 matrix containing the last column of the estimated projection matrix:

$$\mathbf{M}_2 = \mathbf{R} * \mathbf{M}_1,$$

$$\mathbf{m}_2 = -\mathbf{M}_2 * (\mathbf{R} * (-\mathbf{M}_1^{-1} * \mathbf{m}_1) + \mathbf{t}).$$

These formulae will thus allow us to estimate the left projection matrix for any subsequent camera location from the left projection matrix obtained for the first camera location, and similarly for the right projection matrices.

As before, I first prototyped this method in Matlab to see how well it would predict the location of the points. I was able to do this using data from a local test of the photographing procedure, done by other researchers working on the Archeoviz project, which included recording the location of these tripod points, using the same Theodolite method used to record the world location of the markers. The resulting estimation of the projection matrix for the second scene was quite good, with almost all of the markers located within the 30-pixel circle. This was true using only the minimum number of points on the tripod, three. Also, we found that locating the three points on the tripod adds no more than a minute or two to the archaeologists' fieldwork. Thus, this method will allow the process of calibration to be sped up greatly at the cost of a very small increase in the work that the archaeologists in the field must do.

This final method of prediction greatly improves the speed of the calibration process by requiring that users independently locate only six markers in one image of a scene in order to accomplish the calibration for all images in all stereo pairs of that scene. In addition, if the world coordinate system is kept fixed for images of other scenes to be reconstructed, and if the world coordinates of the same, fixed tripod points are again recorded, this method can also be used to estimate the location of the markers in any image of any scene, as long as the location of the markers used in those scenes are recorded relative to this same, fixed world coordinate system. Thus, all of the images for all scenes that need to be reconstructed can be calibrated using only the projection

matrices obtained for the first stereo pair, along with the text files containing the world location of the fixed tripod points.

5. RESULTS AND CONCLUSIONS

Through the development of this calibration implementation, the process of calibrating the cameras has been greatly improved in both the time needed to accomplish it and in the work required by user. This new implementation requires that users find only six markers in one image of a scene by themselves, predicting the location of all of the remaining markers in all remaining images of the scene and of all markers in any images of any scenes in which the world coordinate system is kept fixed, using the three methods of prediction described above. In the original implementation, users needed to find all of 20-25 markers in each of the 6-16 images of a scene with no help at all.

All three of these prediction methods are contained within the same program, and users can utilize any of the three depending upon the situation. Thus, the calibration for all of the images for a given scene can be completed in one use of a single program. This is another improvement over the original method, in which users had to alternate between using Adobe Photoshop or some similar program to locate the markers and record their image coordinates, and Matlab to calculate the projection matrix. Thus, with the new implementation, the time needed to achieve the calibration and the amount of work that users must do independently have been reduced to a very small fraction of those in the original implementation. This allows users to quickly move on to the construction of the three-dimensional models, and thus will more quickly allow them to use this powerful tool for studying the archaeological sites.

6. ACKNOWLEDGMENTS

I would like to thank all of the participants in the Archeoviz project, particularly my advisor, Dr. Kostas Daniilidis, for allowing me to be a part of his research. I would also like to thank Dianna Xu and Sandy Patterson for all of the help that they have given me throughout the summer in my work on the project. In addition, I would like to thank Dr. Jan Van der Spiegel for allowing me to be a part of the SUNFEST program and for organizing the various events and workshops that have made it a unique summer experience. Finally, I would like to thank Microsoft Corporation and the National Science Foundation for providing the funding that has made my summer experience possible.

7. RESOURCES

1. JANUS: A hand-eye humanoid robot: Pinhole camera model, Garcia C., Available from <http://www.csd.uch.gr/~cgarcia/JANUS/pinhole.gif>, accessed 16 July, 2003.
2. D. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Prentice Hall, Upper Saddle River, New Jersey, 2003.

3. CV online: Compendium of computer vision, Editor Robert B. Fisher, Available from: <http://www.dai.ed.ac.uk/CVonline/>, accessed 16 July, 2003.

4. Calculation of the rotation matrix, Young-Hoo Kwon, Available from <http://kwon3d.com/theory/jkinem/rotmat.html>, accessed 22 July, 2003.

5. E. Angel, *Interactive Computer Graphics: A Top-Down Approach with OpenGL*, Addison-Wesley, Reading, Massachusetts, 1997.

APPENDIX A: THE SOFTWARE IMPLEMENTATION

I implemented the final program using Microsoft Visual C++, and used the OpenGL graphics library to achieve the display of the images to be reconstructed. I utilized the GLUT library to handle both the user input, from the keyboard and mouse, and the interface with the operating system to create the windows in which the images are displayed. I also used the GLUI library to create the buttons, text fields, and other controls utilized by the users in the control panel windows.

A screenshot of the first implementation of the program, with only the six-point algorithm used to calibrate for just a single image, is presented in Figure 4 below. The top window on the left is the console window, in which all output from the program will be printed for the user. The main window, labeled “Six Points,” is the window in which the image will be displayed, and the control panel on the right contains all of the buttons and controls that the user will need in order to complete the calibration process.

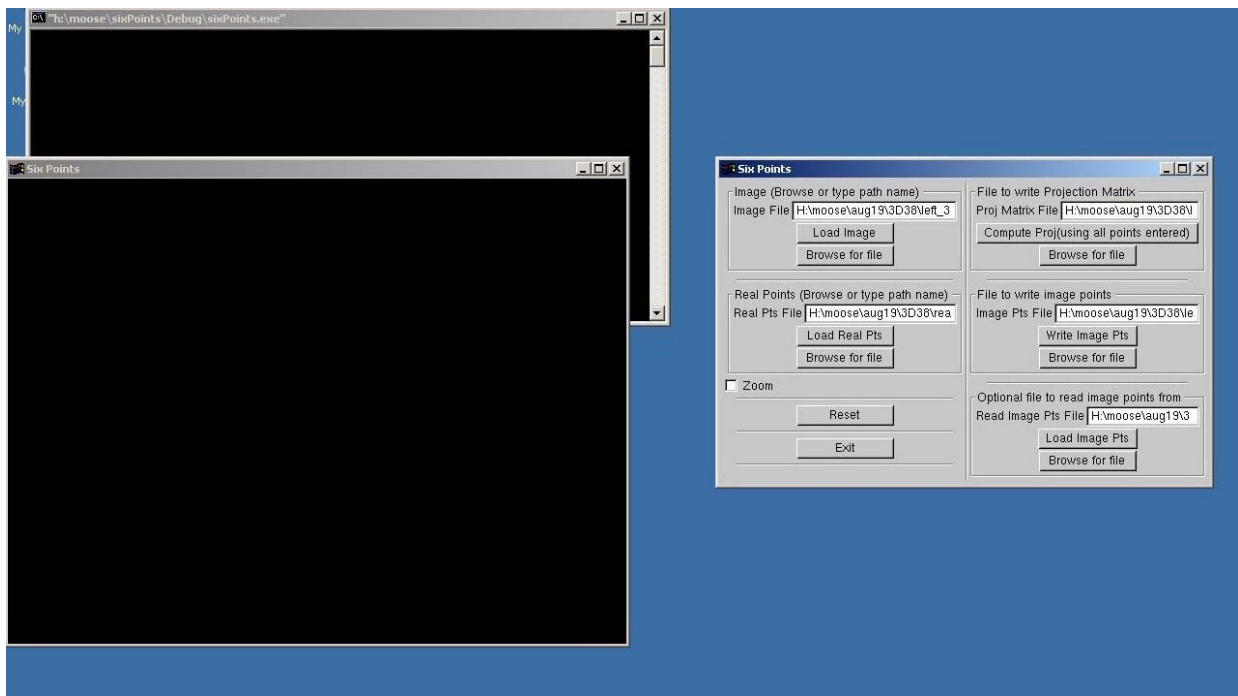


Figure 4: Screenshot of the six points program.

The input to the program is the image, in .tga format, and the text file containing the three-dimensional world coordinates of all of the markers. The output will be two text files, one containing the projection matrix for the scene and one containing the image coordinates for all of the markers in the scene, both of which will be correctly formatted for use in the reconstruction program. The user proceeds by first loading both the image and world coordinate files, and by specifying the two output text files. These files are specified in the text fields in the control panel by either using the browse buttons below the fields or typing in the file names manually. I have implemented the Windows style browsing windows to make the input of files easier, as the text fields are sometimes unstable and it is often tedious to type in the entire pathname of the file. (This problem was also present in the original Archeoviz program for the reconstruction.)

After loading all of the necessary files, the user begins the search for the markers. For the first six markers, the user types in the marker index using the keyboard and then clicks, with the mouse, on its location in the image. The program records the image coordinates of the point selected and places it in the correct location in the output file based on the input index. After these first six points are input, the program predicts the location of the remaining markers, as described above, and draws blue circles around these predicted locations. Figure 5 shows a screenshot of this step in the program, using a sample image from one of the scenes from Tiwanaku, with yellow circles displayed around the six markers entered.

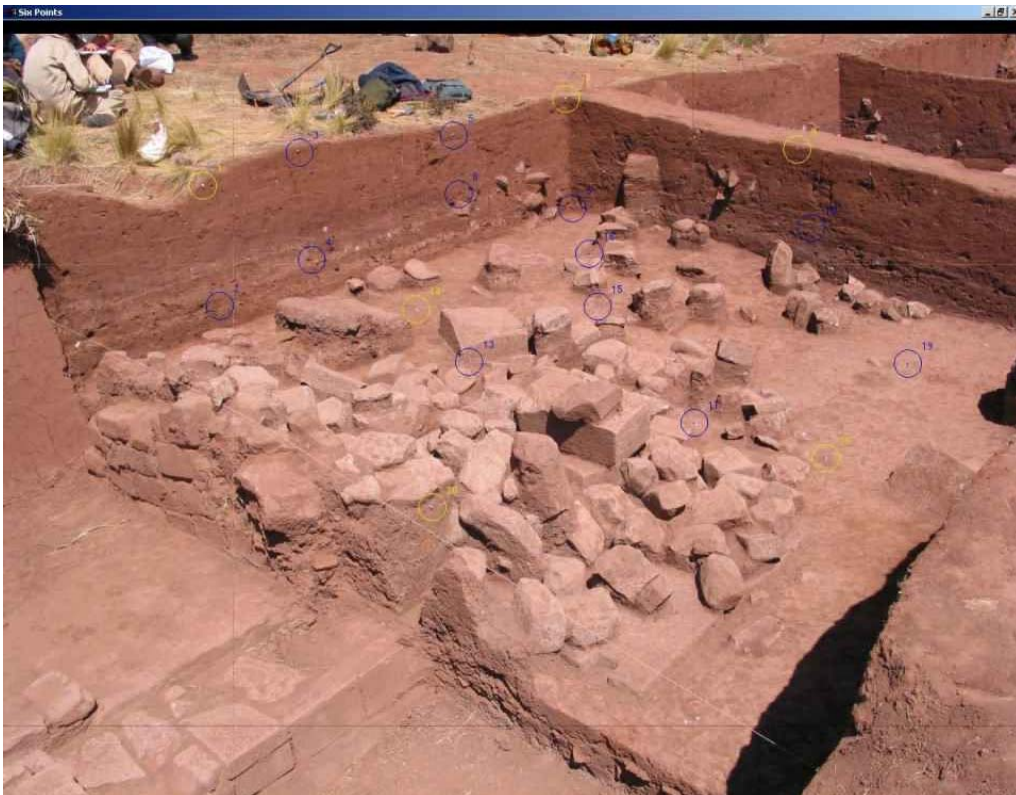


Figure 5: A screenshot of the prediction step in the program. The yellow circles indicate the location of the first six markers, and the blue circles the predicted location of the remaining markers.

Next, the display will zoom to each of these predicted locations, one by one. When zoomed on a circle, the user will simply need to click on the appropriate location on the screen and the program will record the image coordinates, automatically identifying the correct index for the marker. If the prediction of a particular marker is not accurate, the user can simply type the “s” key to skip it and enter it later. After all of the markers have been entered or skipped, the display returns to the original view. If any more markers need to be entered, the user can enter them in the same manner as the first six markers. Alternatively, at any point after the first six markers are entered, the user can click on the “Compute Proj.” button on the control panel to get a better estimate of the projection matrix using all of the point correspondences entered, and thus get a better estimate for the location of the points not yet entered. If the marker is within the new prediction circle, the user then can simply click on it, without worrying about entering the index of the marker, which will be displayed above the circle. After all of the markers in the scene have been clicked and their image coordinates entered, the program will calculate a projection matrix using the correspondences from all markers and will output both this matrix and the image coordinates to the appropriate text files. The user can then take these text files immediately to the reconstruction program and begin building the three-dimensional models.

I have implemented the method of estimating the right projection matrix from the left as an extension to the original six points program described above. I have created a second window and control panel for the right image, identical to the window and control panel used for the left. In addition, I have moved all of the variables and controls that relate to both images, such as the real points file and the controls that specify d and θ , to the main control panel at the top of the screen. Figure 6 shows a screenshot of the extended version of the program.

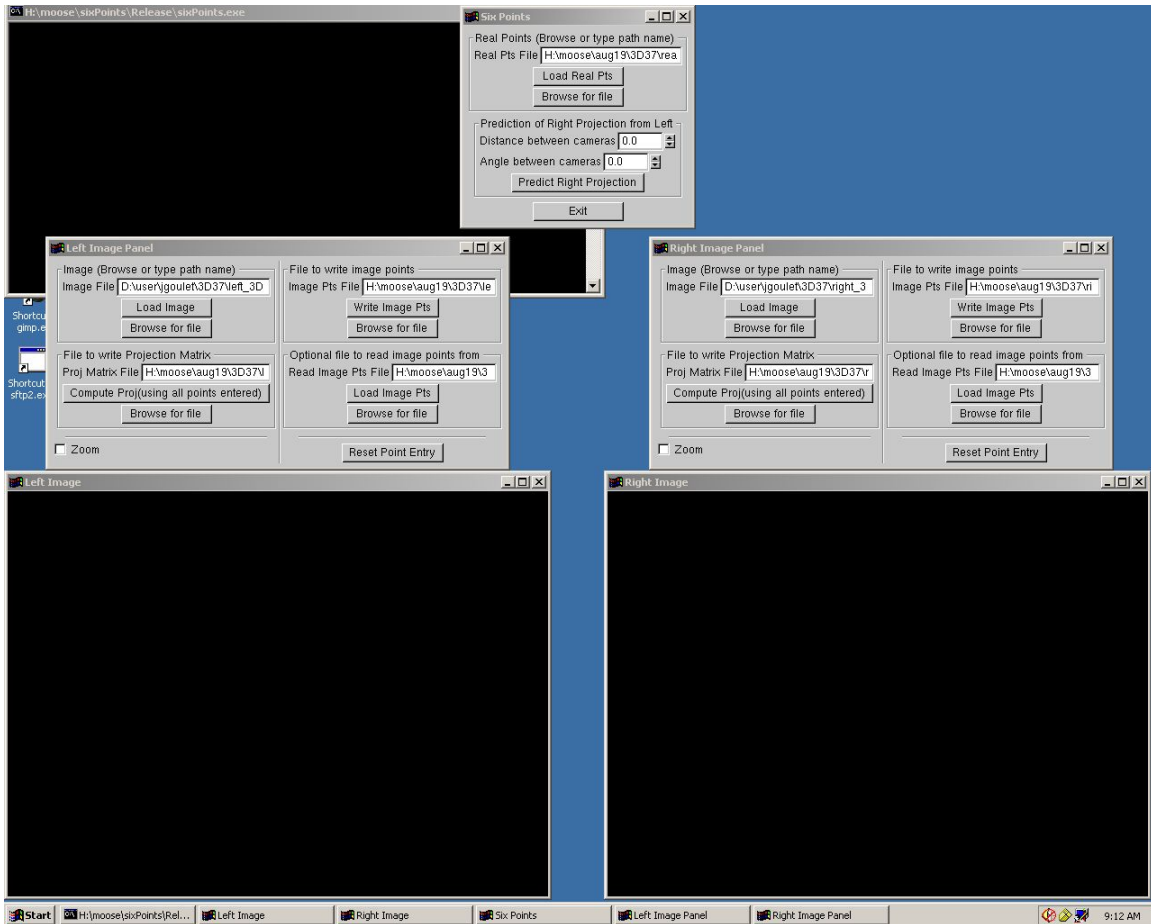


Figure 6: Screenshot of the extended version of the calibration program.

In the extended program, the user accomplishes the calibration of the left image exactly as in the original implementation of the program. Then, the user selects the appropriate values for d and θ , using the “spinners” in the main control panel, and clicks on the “Predict Right Projection Matrix” button. The program then extracts and stores the intrinsic and extrinsic parameters from the left projection matrix, using various formulae for calculating these parameters from the rows of the projection matrix. It calculates the transformation from left to right camera coordinate systems using the selected parameters and combines this transformation with the extrinsic parameters from the left projection to get an estimate for the extrinsic parameters of the right image. It uses these parameters, setting the right intrinsic parameters to be the same as those of the left image, to compute the projection matrix for the right image and then uses this to predict the location of all of the markers in the right image. The user then enters the image coordinates for the markers in the right image in exactly the same way as done for the predicted points in the left. After all markers have been selected in the right image, the image points and projection matrix text files are output exactly as they had been for the left image, in the same manner as in the original implementation of the program.

I have incorporated the final method of prediction, using the tripod data, as another extension to my program through adding four more text fields to the main control panel. These fields will be used to specify four additional text files. The first pair will be the text files containing the projection matrices for the left and right images in the first stereo pair, obtained using the program in the original manner. The second pair will contain the world coordinates of the points on the tripod, one at the location of the first stereo pair and the second at the subsequent location for which the user is currently calibrating. Figure 7 shows a screenshot of the new main control panel.

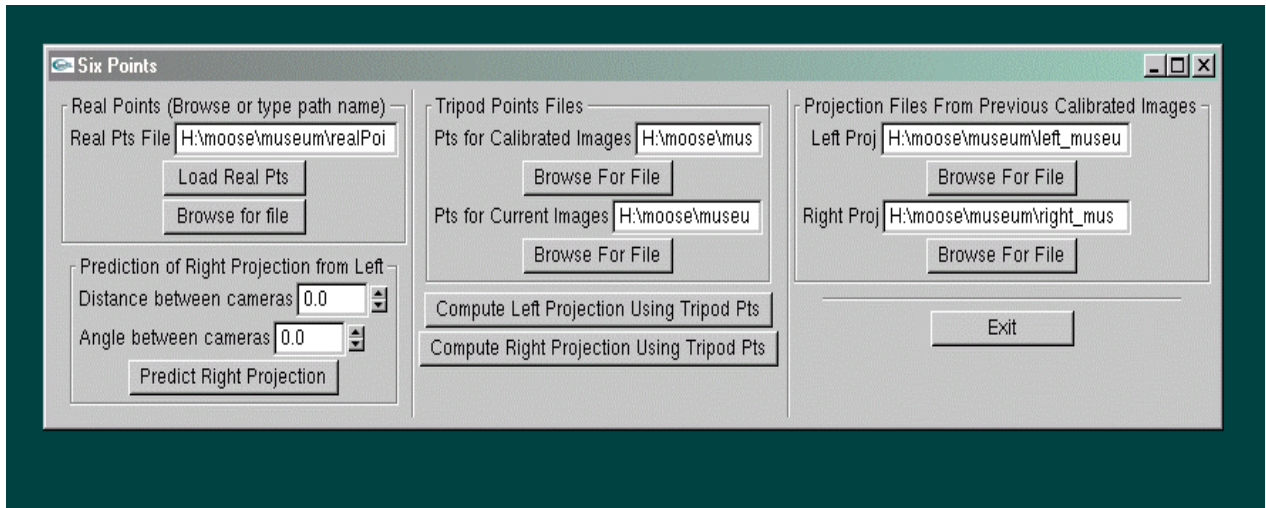


Figure 7: The new control panel for the final extension to the program.

The user specifies these four files, using the same browsing windows as before, and then clicks a button to calculate the projection matrices for the current images, using the above method for rigid transformations. The program will then predict the location of all points in both images in the current stereo pair, without requiring the user to locate any of the markers. The user can then utilize these predictions to input the image coordinates of the markers exactly as done for all of the predicted locations in the previous versions of the program. The output will again be the same projection and image point files for both the left and right images in the stereo pair, the first of which can be used in future scenes as the input projection matrices used in the tripod method.

This implementation includes various features that make it easy to use and user-friendly for the archaeologists in the field. The friendly graphical interface and easy-to-use zooming system allow the user to quickly learn how to use the software. In addition, the browsing windows provide a quick and easy method for loading all necessary files and for saving all data obtained. Also, the console provides the user with valuable information, including which markers are entered as they are clicked, how many markers remain to be entered, and tips on using some of the features of the program, such as the zooming buttons. In addition, the program automatically keeps track of the indices of the markers and displays them for the user above the circles around the markers. Finally, it automatically outputs all necessary files, properly formatted, to be used immediately in

the reconstruction. All of these features help to improve the performance of the program and add to the effectiveness of the underlying algorithms in speeding up the process of calibration and allowing the user to quickly proceed to creating these valuable graphical models.