A Novel Camera Calibration Algorithm as Part of an HCI System: Experimental Procedure and Results

K. Sauer, E. Yfantis, M. Teruel, R. El-Khater
Computer Graphics and Image Processing Laboratory
School of Computer Science, University of Nevada, Las Vegas
Las Vegas, NV, USA 89154, USA

ABSTRACT

Camera calibration is an initial step employed in many computer vision applications for the estimation of camera parameters. Along with images of an arbitrary scene, these parameters allow for inference of the scene's metric information. This is a primary reason for camera calibration's significance to computer vision. In this paper, we present a novel approach to solving the camera calibration problem. The method was developed as part of a Human Computer Interaction (HCI) System for the NASA Virtual GloveBox (VGX) Project. Our algorithm is based on the geometric properties of perspective projections and provides a closed form solution for the camera parameters. Its accuracy is evaluated in the context of the NASA VGX, and the results indicate that our algorithm achieves accuracy similar to other calibration methods which are characterized by greater complexity and computational cost. Because of its reliability and wide variety of potential applications, we are confident that our calibration algorithm will be of interest to many.

Keywords: Camera Calibration, Camera Modeling, Lens Distortion, 3D Reconstruction

1. INTRODUCTION

Camera calibration is a useful tool in the fields of computer vision, computer graphics, robot vision, etc. In fact, the tasks common to these various fields require an understanding of the 3D structure of a scene based on the information extracted from a sequence of images taken by two or more cameras. The process of translating the images into information that can be used to reconstruct the scene observed involves several steps, one of the most important being that of camera calibration. Calibrating the cameras is equivalent to initializing the system, given that we obtain optimal values for the camera parameters. These parameters allow us to extract metric information from the scene and thereby locate objects in space using the 3D structure recovered from the 2D images.

The calibration algorithm we will present in this paper has been developed as part of the research underway at the Computer Graphics and Image Processing Laboratory at the University of Nevada, Las Vegas. This research project is supported by NASA[1], and its main goal is to produce an efficiently effective application for Human- Computer Interaction (HCI) that is, most importantly, *non-invasive*. Thus, we are developing a system in which the actor is unhampered by any kind of external device[2]. Specifically, we will determine the motion of the hand solely from images tracking the movement of the hand where the subject wears, at most, a pair of ordinary gloves with colored markers at key locations. This will allow us to gather the information required to accomplish our task of resolving the hand's 3D coordinates, thereby estimating its pose and motion.

The structure of this paper is as follows: Section 2 contains a brief description of our algorithm. The reader is introduced to our experimental tools and procedures in Section 3. Section 4 provides the practicum and experimental results. Finally, we present our conclusions in Section 5.

2. CAMERA CALIBRATION ALGORITHM

We chose the Pinhole Camera Model for our project. This model is based on the principle of collinearity, where each point in the space is projected by a straight line through the projection center into the image plane [1].

As mentioned above, camera calibration allows for the estimation of all parameters necessary for obtaining the correspondence between image coordinates (2D) and world coordinates (3D) [3], [4], [5]. The parameters can be divided into two categories:

1) *Intrinsic Parameters*, which provide a link between the Pixel Coordinate System (PCS) and the Camera Coordinate System (CCS), and
2) *Extrinsic Parameters*, which define the location and orientation of the camera with respect to the World Coordinate System (WCS).

2.1 Estimation of Internal Camera Parameters

The internal parameters to be estimated are the focal length f, the scale factors S_x and S_y (in the x and y directions, respectively), and the radial distortion coefficients k_0 and k_1. In order to calculate the internal parameters, we place our calibration object perpendicular to the camera, and we set the origin of the WCS to be the point projected onto the center of the image, thus setting the rotation matrix $\boldsymbol{R}$ to a 180 degree y-

[1] NASA Space Grant/EPSCoR: "Development of a Nationally Competitive Program in Computer Vision Technologies for Effective Human-Computer Interaction in Virtual Environments"

[2] For more information refer to http://biovis.arc.nasa.gov/vislab/vgx.htm

roll and the translation vector $\mathbf{t}$ to a zero vector of the form $(0, 0, h)$.

Given that the ratio between the length of a line in the real world and the length of that line in the image remains constant for all lines (Fig. 1), we can estimate the focal length as follows: refer to Fig. 1 and consider the projection of the two lines x_1x_2 and x_3x_4, where the latter is raised to a known height z_3. Since $\Delta px_1x_2 \approx \Delta pu_1u_2$ and $\Delta px_3x_4 \approx \Delta pu_3u_4$, we have that

$$\frac{S_x \left| u_2 - u_1 \right|}{\left| x_2 - x_1 \right|} = \frac{f}{h} \qquad \textbf{(1)}$$

and

$$\frac{S_x \left| u_4 - u_3 \right|}{\left| x_4 - x_3 \right|} = \frac{f}{h - z_3}. \qquad \textbf{(2)}$$

Also, since it is known that there is a constant ratio between f and S_x and that both parameters have the same effect on the image (scaling), we can define a "Virtual Camera" in which we set S_x to one pixel per centimeter and obtain a corresponding focal length f. We can thereby eliminate S_x from Eqs. (1) and (2), which allows us to estimate f.

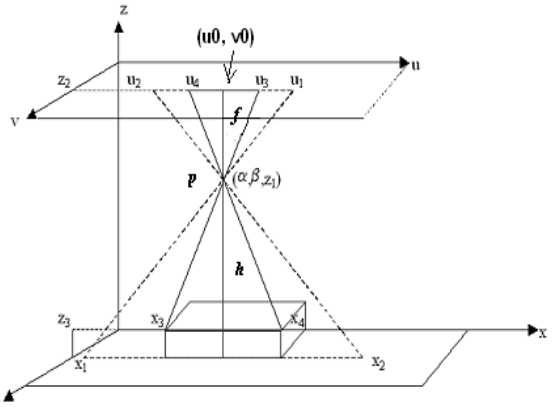

Fig. 1: Finding the focal length from two lines at different heights

To best estimate f, S_x and S_y, given that Eqs. (1) and (2) do not model radial distortion, we must consider lines that lie entirely within a region nearby the center of the image since this area is least affected by radial distortion.

Let us note that in many cameras, S_x and S_y are not equal; as a result, the pixels have a rectangular shape. The ratio S_x / S_y is called the *aspect ratio* and will be denoted by ρ [8].

As we just mentioned, f, S_x, and S_y are determined by using only control points lying close to the center of the image where the effects of radial distortion are minimal. However, when trying to arrive at a viable estimate of the distortion coefficients, the control points to be considered must be uniformly scattered throughout the entire image. The inclusion of lens distortion is extremely important for the accuracy of our model since lens distortion has a significant impact on the shape of the image. The most commonly considered form of distortion is radial, which is due to imperfections in the construction of the lens and causes points to be displaced radially from the center of the image. It can be shown that the formulas modeling radial distortion are given by

$$\begin{bmatrix} \delta_x^{\mathrm{r}} \\ \delta_y^{\mathrm{r}} \end{bmatrix} = \begin{bmatrix} \tilde{x}\left(k_0 r^2 + k_1 r^4\right) \\ \tilde{y}\left(k_0 r^2 + k_1 r^4\right) \end{bmatrix} \qquad \textbf{(3)}$$

where $r^2 = \tilde{x}^2 + \tilde{y}^2$ [2]. Then, once we include Eq. (3) in our model, we can solve for the distortion coefficients k_0 and k_1 via a Linear Least Squares (LLS) Fit [8].

2.2 Estimation of External Camera Parameters

In this section, we determine the orientation of the camera with respect to the WCS. Our goal is to estimate both the rotation with respect to each of the axes in the WCS and the total translation.

Initially assuming that the camera is only rotated about the *x*-axis by an angle of θ, the rotation matrix is an *x*-roll (Fig. 2). Thus, from Fig. 2 it follows that

$$\frac{ED}{a} = \frac{f - ED \cdot \sin\theta}{h} \qquad \textbf{(4)}$$

and

$$\frac{\lambda ED}{a} = \frac{f - \lambda ED \cdot \sin\theta}{h}. \qquad \textbf{(5)}$$

Thus, θ may be expressed as

$$\theta = \sin^{-1}\left(\frac{h\left(\lambda - 1\right)}{a\left(\lambda + 1\right)}\right). \qquad \textbf{(6)}$$

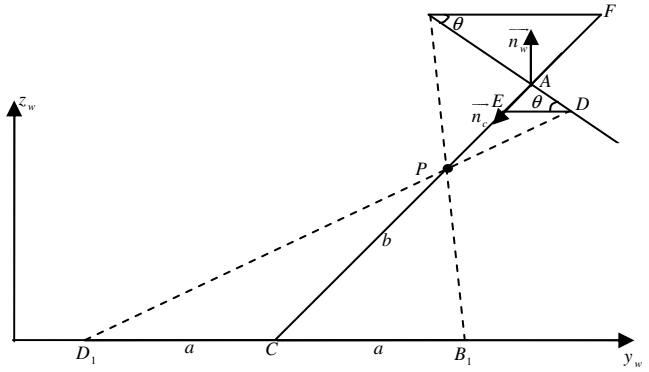

Fig. 2: Rotation of the camera by θ degrees about the *x*-axis

Refer to Fig. 2, where $\mathbf{n}_c$ is the normal to the image plane, $\mathbf{n}_w = (0,0,1)^T$ is the normal to the $x_w y_w$ plane, and $\pi + \theta$ is the angle between them. Therefore, if $\boldsymbol{R}_x(\theta)$ is a rotation matrix about the x-axis, it is clear that

$$\mathbf{n}_c = \boldsymbol{R}_x(\pi + \theta)\mathbf{n}_w . \quad \textbf{(7)}$$

Thus, we are able to find $\mathbf{n}_c$ for the case in which the camera is only rotated about the x-axis.

In general, however, the image plane is not rotated solely about a single axis. As a matter of fact, we will usually have a sequence of rotations about the *x*, *y*, and *z* axes. Hence, in order to find the angle between $\mathbf{n}_c$ and $\mathbf{n}_w$, we must first identify an axis of rotation in the $x_w y_w$ plane. In order to find the axis of rotation, we rely on the fact that the image of the axis of rotation remains unchanged as the camera is rotated. Once the axis is identified, we can use Eq. (6) to compute the angle of rotation. To this end, we constructed a special calibration object that consists of points in a circular pattern which lies centered at the origin of the WCS. We search for an opposing pair of points that are equidistant from the center of the circle. Subsequently, these points are used to determine the axis of rotation, which is in fact the line connecting them. Next, we use the line perpendicular to the axis with Eq. (6) to determine θ.

Once the angle of rotation θ has been found, we can proceed with the construction of the corresponding rotation matrix, which will be denoted by $\boldsymbol{R}_{\vec{u}}(\theta)$. This allows us to find $\mathbf{n}_c$ and the coordinates of the center of the image plane, C. Hence, we can compute the equation of the image plane, which enables us to construct the sought-after rotation matrix, $\boldsymbol{R} = \begin{bmatrix} \mathbf{du} & \mathbf{dv} & \mathbf{n}_c \end{bmatrix}$ [8].

3. EXPERIMENTAL TOOLS AND PROCEDURES

3.1 Experimental Tools

The experimental tools developed for this project fall into two categories: physical tools and software tools. The physical tools include a circular pattern, a grid pattern, and a precisely measured box (the NASA GloveBox). The software tools consist of a Region Selection Tool (RST), a Dot Extraction Tool (DET), and tools for rearranging extracted points.

The circular pattern (Fig. 3) is used to carry out the implicit calibration, as described in Section 2.2. It consists of 180 evenly spaced dots that form a circle of radius 10 in. Each of the constituent dots is a circle of radius 0.04 in. The grid pattern (Fig. 4) is the one used for finding the internal parameters of the camera (Section 2.1) as well as testing the final calibration. The grid consists of 0.04 in.-radius dots that are drawn in an 11x14 grid pattern, with neighboring dots spaced 1 in. apart.

The RST is necessary to select the calibration pattern in the image to restrict the search for dots to that region. This tool works by enclosing the region of interest within a closed polygon, specified by the user, so that the DET only considers points that fall within this polygon. It is important to note that the polygon should enclose the entire calibration pattern.

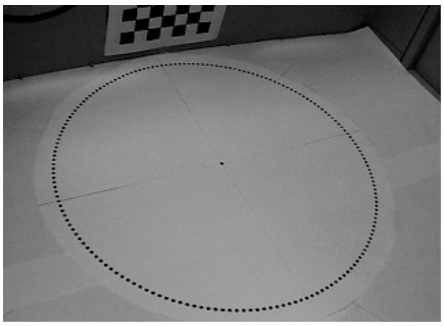

Fig. 3: Circular Calibration Pattern

Once the region has been selected, the DET is used to accurately detect the pixel coordinates of the control points on the calibration patterns. To accomplish this, the image is converted to YIQ and then segmented into black and white regions via a thresholding. Finally, the center of mass of each of the dot-representative black regions is computed via a Depth First Search (DFS) sweep.

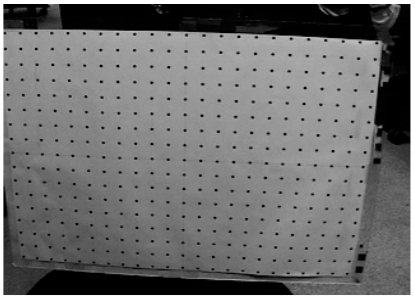

Fig. 4: Calibration Grid

Next, we utilize our point ordering tools to produce a meaningful arrangement of the control points. We have developed two tools for this purpose: a tool for arranging the points from the circular pattern into clockwise order, and a tool for ordering the points in the grid line by line.

1) The circle ordering tool works by separating the points into four regions in the manner shown in Fig. 5. Due to the fact that the image's origin is located at its top left corner, the points in Region 1 are sorted by increasing y coordinate; in Region 2 they are sorted by decreasing x; in Region 3 by decreasing y, and in Region 4 by increasing x. This results in a clockwise ordering of the points.

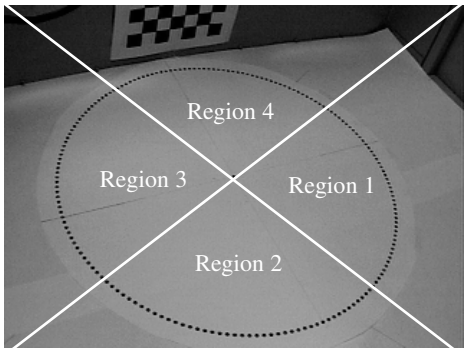

Fig. 5: Regions used for ordering the dots

2) The tool for ordering the control points on the grid sorts the points line by line from left to right (Fig. 6) as follows: the four corners of the grid are the algorithm's input. Let $\{c_{bl}, c_{tl}, c_{br}, c_{tr}\}$ be the corners, where c_{bl} is the bottom left corner, c_{tl} is the top left corner, c_{br} is the bottom right corner, and c_{tr} is the top right corner (see Fig. 6). The initial step is to find the head and tail of every line. This is accomplished by first computing the slope s_1 of the line $c_{bl}c_{tl}$. For all remaining points p_i, we find the slope of the line $p_i c_{tl}$ or $p_i c_{bl}$; if it is equal to s_1, then p_i is considered a head of a line. Similarly, using $c_{br}c_{tr}$, we can find all the tails of the lines.

Once we have found heads and tails, we sort each set in increasing order according to the y-coordinate. The sorting is done in a manner similar to that described above. In fact, we consider every line formed by joining the corresponding head and tail. That is, we consider every line l_i formed by heads[i] and tails[i]. Let s_i be the slope of this line. For all the remaining points p_j, if the slope of the line p_j heads[i] or p_j tails[i] equals s_i, then p_j belongs to the line l_i. Then, we sort all the points in every line l_i in increasing distance from heads[i]. The sequence of lines $l_1, l_2, l_3 ..., l_N$ gives us the desired ordering of the points.

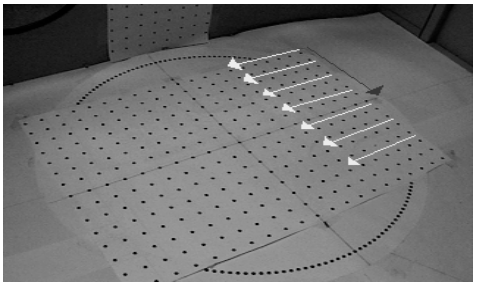

Fig. 6: Arrows to indicate sorting direction

3.2 Experimental Procedure

We begin by finding the internal camera parameters as described in Section 2.1. The camera is placed perpendicular to the grid as shown in Fig. 1, and an image of the grid is captured. We extract the 2D image coordinates of this grid using the DET and store them in the ordered set LP. Next, we raise the grid using a precisely measured height and fill the ordered set HP. The sets LP and HP are rearranged using the point ordering tool described in the previous section and then used with the solution of Eqs. (1) and (2) to compute the focal length f for the virtual camera.

In the next step, we compute the radial distortion coefficients by projecting all the points in LP from the 3D WCS to the 2D PCS using the values for f and the height of the camera h just computed. The actual image coordinates of the points are the ones that were extracted from the image using the DET; these are used to find the distortion at each point. Then, we fit a polynomial having a form akin to that of Eq. (3) to the plot of distortions using LLS, yielding the values of the radial distortion coefficients k_0 and k_1.

Once we have obtained the internal camera parameters, we can proceed with the external calibration to determine the camera's rotation and translation. To do so, we place the camera at the desired location, and then we position the circular pattern described in Section 3.1 so that the center of the circle appears at the center of the image. The dot extraction tool is used to extract the ICS coordinates of the dots and store them in the ordered set CP; then, the circle ordering tool is used to order the points in CP. We then proceed to determine the axis of rotation of the camera as follows: every two points on opposite sides of the circle are paired together into pairs (p_{i1}, p_{i2}), where i is the pair number and $p_{i1} = (c_{i1}, r_{i1})$ and $p_{i2} = (c_{i2}, r_{i2})$ are the image coordinates of the two points. For every pair (p_{i1}, p_{i2}), we compute the ratio $r_i = \dfrac{\|p_{i1} - c\|}{\|p_{i2} - c\|}$, where $c = (c_c, r_c)$ represents the image coordinates of the center of the circle. The line corresponding to the pair that gives r_i closest to one is taken to be the axis of rotation, which we denote by a vector $\vec{v}$. We take the two circle points that fall on the line that is perpendicular to $\vec{v}$ and compute the angle of rotation θ about $\vec{v}$ using Eq. (6). Once we have θ, we can compute the

rotation matrix of the camera, $\boldsymbol{R} = \left[\mathbf{du} \quad \mathbf{dv} \quad \mathbf{n}_c \right]$, as mentioned in Section 2.

4. PRACTICUM

A set of known control points was used to verify the correctness of the calibration. The set contains 294 control points, positioned every 1 in. in a 21 by 14 grid pattern. First, an image of the control points is captured from each calibrated camera, and the points are projected from the PCS to WCS using the parameters computed from the calibration procedure. The Euclidean distance between the computed WCS coordinates and the actual known world coordinates is taken as the error measure.

Two types of errors are considered, the calibration error and the reconstruction error. For the calibration error, the points are assumed to belong to a known plane, say the $z = 0$ plane, and are projected using only one camera. Before projecting the points, we compensate for the radial distortion, as mentioned in Section 3.2. Next, the points are projected first into the CCS and then into the WCS using the estimated external parameters. To compute the reconstruction error, images from the two cameras are used in conjunction to resolve all three coordinates simultaneously via triangulation.

Table 1: Average Calibration Errors

	ERRORS (cm)			
Trial No.	X	Y	Z	Magnitude
1 – Cam 1	0.085	0.155	0.158	0.237
1 – Cam 2	0.127	0.106	0.065	0.178
2 – Cam 1	0.071	0.120	0.071	0.156
2 – Cam 2	0.168	0.214	0.177	0.324
3 – Cam 1	0.195	0.140	0.141	0.278
3 – Cam 2	0.100	0.185	0.160	0.264
4 – Cam 1	0.093	0.246	0.231	0.350
4 – Cam 2	0.112	0.104	0.042	0.158
5 – Cam 1	0.107	0.057	0.106	0.161
5 – Cam 2	0.250	0.179	0.150	0.342
6 – Cam 1	0.111	0.089	0.114	0.182
6 – Cam 2	0.187	0.069	0.149	0.248
7 – Cam 1	0.132	0.189	0.082	0.244
7 – Cam 2	0.196	0.132	0.199	0.254
8 – Cam 1*	0.152	0.178	0.210	0.314
8 – Cam 2*	0.161	0.083	0.085	0.200
9 – Cam 1*	0.060	0.082	0.056	0.116
9 – Cam 2*	0.091	0.175	0.121	0.231
10 – Cam 1*	0.114	0.288	0.207	0.372
10 – Cam 2*	0.081	0.124	0.097	0.177
Average Errors	0.1296	0.1457	0.1310	0.2393

* Trial carried out with the cameras closer together.

Table 2: Average Reconstruction Errors

	ERRORS (cm)			
Trial No.	X	Y	Z	Magnitude
1	0.086	0.285	0.197	0.357
2	0.145	0.138	0.161	0.257
3	0.101	0.185	0.512	0.553
4	0.109	0.169	0.160	0.256
5	0.194	0.188	0.265	0.378
6	0.150	0.139	0.189	0.278
7	0.131	0.107	0.251	0.279
8*	0.181	0.293	0.355	0.398
9*	0.113	0.197	0.230	0.323
10*	0.098	0.202	0.190	0.294
Average Errors	0.1308	0.1903	0.2510	0.3373

As shown by Tables 1 and 2, we obtain an error of about 3 mm in the 3D coordinate reconstruction. Some of the possible sources of error in our calibration are measurement error, feature extraction, triangulation error, and other types of lens distortions. In fact, the feature extraction accuracy can be affected by non-uniform illumination, which causes the detected centroids of the control points to be slightly displaced from the actual centers. Regarding the lens distortion, as mentioned in Section 3, our calibration model only considers radial lens distortion. However, considering other forms of distortion such as tangential and thin prism may improve the results of the calibration. Finally, the triangulation stage contributes to the overall error due to the fact that the accuracy of triangulation depends not only on the baseline (distance between the cameras) but also on the angle between the rays (Fig. 7). In fact, points are less precisely localized along the ray of projection as the rays become increasingly parallel. In the NASA VGX, the setup requires that the cameras be close to each other; as a consequence, the rays tend to be nearly parallel. This impacts our error, which can be seen in the reconstruction errors in Table 2.

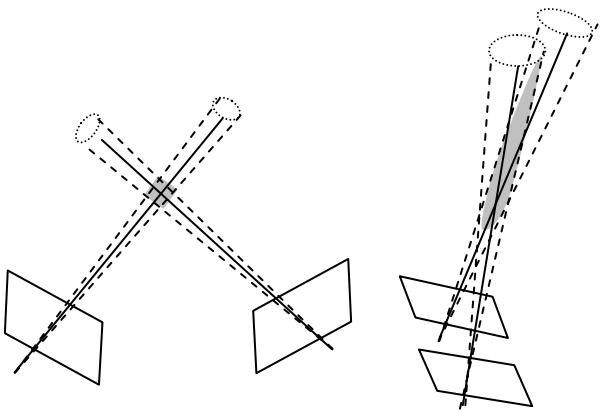

Fig. 7: The shaded region represents the reconstruction uncertainty region, which depends on the angle between the rays of projection and increases as the cameras are placed closer together.

5. CONCLUSIONS

We have presented a novel approach to camera calibration based on the geometric properties of the camera and system. Our calibration algorithm has various advantages and disadvantages in comparison to other calibration schemes. For instance, it is computationally fast and algorithmically straightforward. In fact, our algorithm yields a closed form solution without relying on non-linear optimization methods. However, two separate procedures are needed for computing the camera's intrinsic and extrinsic parameters. Yet, once the internal camera parameters are computed, it is possible to reposition the camera without recomputing them. As mentioned earlier, the accuracy of our calibration method could be further improved by considering other forms of lens distortion such as tangential and thin prism distortion. It would also be possible to use the parameters obtained from our calibration method as an initial guess to some of the non-linear optimization methods used in the literature [4], [5], [9]. Finally, we would like to remark that the calibration method presented in this paper was successfully used to calibrate the cameras in the NASA VGX. Using the calibrated cameras, we were able to extract the positions and directions of the hands in the VGX along with the positions and directions of each individual finger with high accuracy [6]. This allowed us to reconstruct the world coordinates of key locations on the user's hand that were fed into our 3D hand modeling module, yielding a precise and realistic virtual modeling of the human hand [7].

6. REFERENCES

[1] O. Faugeras, **Three-Dimensional Computer Vision: a Geometric Viewpoint**, Cambridge, MA: MIT Press, 1993.

[2] J. Weng, P. Cohen, and M. Herniou, "Camera Calibration with Distortion Models and Accuracy Evaluation," **IEEE Trans. Pattern Analysis and Machine Intelligence**, Vol. 14, No. 10, 1992, pp. 965-980.

[3] M. B. Teruel, R.W. El-Khater, O. Kubushyna and E. A. Yfantis, "A Review Paper on Camera Calibration Techniques," **Technical Report, NASA GloveBoX Project**, Computer Graphics and Image Processing Laboratory, School of Computer Science, University of Nevada, Las Vegas, NV, 2003.

[4] O.D. Faugeras, G. Toscani, "The Calibration Problem for Stereo," **Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)**, 1986, pp. 15-20.

[5] O.D. Faugeras, G. Toscani, "Camera Calibration for 3D Computer Vision," **Proc. Int'l Workshop Industrial Applications of Machine Vision and Machine Intelligence**, 1987, pp. 240-247.

[6] M.B. Teruel, O. Kubushyna, R.W. El-Khater, E.A. Yfantis, R. Boyle, "On Estimation of 3D Hand Position Using Stereo Vision," **Proc. of the ISCA 19th International Conference on Computer and their Applications**, 2004, pp. 47-50.

[7] O. Kubushyna, E.A. Yfantis, L. E. Teruel, C. Hwang, P. Stubbedurd, G. Bebis, R. Boyle, **"**Interactive 3D Hand Model with Constraints," **Proc. of the ISCA 12th Int'l Conference on Intelligent and Adaptive Systems and Software Engineering**, 2003, pp. 162-165.

[8] R. W. El-Khater, **"**Camera Calibration for the NASA Virtual Glovebox Project," **MS Thesis, School of Computer Science, University of Nevada, Las Vegas**, Sept. 2004.

[9] J. Weng, P. Cohen and M. Herniou, "Camera Calibration with Distortion Models and Accuracy Evaluation", **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Vol. 14, No. 10, 1992, pp 965-980.