

Automatic Training of a Neural Net for Active Stereo 3D Reconstruction

J. Neubert, T. Hammond, N. Guse, Y. Do⁺, Y. Hu and N. Ferrier*
Mechanical Engineering, University of Wisconsin, Madison, WI, USA 53706
⁺Taegu University, Kyungsan-City, Kyungpook, 712-714, Korea
neubert@robios6.me.wisc.edu ferrier@robios6.me.wisc.edu

Abstract

This paper addresses the problem of recovering 3D geometry using an active stereo vision system. Calibration procedures can be adapted to the active stereo configuration, however, considerable effort is required to accurately model and calibrate the kinematics to avoid poor reconstruction. In the active stereo case there will also be errors due to uncertainty in the kinematics of the system. In addition, data collection needs to be automated because active stereo requires significantly more information for calibration. We present a biologically inspired neural network trained to determine the mapping between 3D geometry and stereo image points. To train the network, we have developed a system to automatically collect accurate calibration data. We compare the reconstructed 3D geometry obtained using a kinematic model based approach with our neural network approach.

Keywords: Neural Nets, Active Vision, Stereo Vision Calibration, 3D Reconstruction

1 Introduction

Obtaining accurate three dimensional (3D) positional information is of great importance in many robotic applications. An effective approach is to use stereo cameras to infer the 3D position from stereo images. For this purpose when using classical stereo approaches, two critical problems must be solved: camera calibration and the stereo feature matching (correspondence problem). While much of the research has been focused on the latter problem, camera calibration is a prerequisite of most vision-related computational problems including stereo matching [5].

The classical camera calibration problem consists of two sub-problems: data collection and camera modeling. A popular method to collect data is to use a flat

panel printed with accurate calibration patterns. For example, Weng et al. [15] used diamond-shaped patterns printed on an ultra flat optical glass by means of a high-precision photographic process that kept the positional error of the pattern within 0.05mm. Accurate world coordinates can be obtained by positioning this panel in front of the camera using high precision positioning such as a robot [10], a linear slide [1], or micrometer [14]. The corresponding image locations of the pattern points must also be determined accurately through appropriate image processing such as corner detection or intersection of the edges [15]. Finally, 3D points and their corresponding image points must be paired. Often this final step is performed with manual intervention to ensure accurate correspondence, which is tedious especially for active head calibration requiring correspondence for multiple images. Developing and utilizing high-precision calibration equipment is very costly and may not be practical in many applications. In addition, calibration of *active stereo heads* cannot follow this procedure directly in that the pose of the cameras changes continuously. Methods must be adapted to account for the motion of the stereo cameras.

During the calibration procedure the collected data, 2D - 3D corresponding points, are used to fit parameters to an assumed camera model that represents the mapping from 3D to the image plane. Because the optical characteristics of lenses differ from camera to camera it would be difficult for any specific camera model to out perform all others. Most existing camera calibration techniques are based on the pin-hole camera model [16] that describes a linear projective mapping between 3D points and their corresponding 2D projection. While the perfect pin-hole camera model is fast and simple, it fails to model non-linearity due to lens distortion. Non-linear models have been proposed to correct radial [14] and other types of lens distortion [10, 5, 16], however, it has been reported that

*This research supported in part by NSF IRI-9703352.

these models may yield worse results than the simple pin-hole model when lens distortion is relatively small [12]. Moreover, sophisticated models generally require more complex and highly accurate calibration. As such, they will be expensive and may not be practical. Thus, a camera model must account for the characteristics of the lens and be computed using accuracies available from the calibration equipment.

In this paper we address both the data collection and camera modeling issues in order to obtain accurate 3D information from an active stereo head. We postulate that using a neural network (NN) may be an attractive alternative to explicit camera models. There are currently two approaches using NNs for camera calibration: using a NN to correct for errors in the pin-hole model, e.g. to model the lens distortion, or, employing a NN to learn the mapping from stereo image points to 3D coordinates [7]. Our approach is the latter. We compare the performance of the neural net to stereo reconstruction methods (when relative camera positions are known) in computing 3D coordinates from stereo image pairs and stereo head pose.

We automated the data collection procedure using two cameras mounted on computer-controlled pan-tilt units and a LCD video projector. Controlled positioning of visual stimuli using the projector with synchronized acquisition of the image points allowed us to perform accurate correspondence automatically. In this manner large amounts of data can be collected for accurate NN training. Similar work by Angrilli *et al.* [1] utilizes a linear slide to present printed patterns to fixed stereo vision system and collect data for NN training. Our work differs in that we have an active robot head, and our NN has a different structure. As discussed below the addition of motion to the stereo system requires significantly more data for the calibration processes.

In the remainder of this paper, we briefly review calibration techniques for active stereo heads. Then we outline our data collection procedure and present a biologically inspired NN to model the stereo to world coordinate mapping. We compare our NN model results to that obtained using a calibrated stereo system (based on calibration developed by Tsai [14] with an extension for active heads developed by Li [9]).

1.1 Calibration of Stereo Heads

Calibration of an *active* stereo head is difficult in that the extrinsic parameters change continuously. In the active vision paradigm, the camera pose is controlled and the extrinsic parameters can be parameterized by the motion parameters (motion of the “eyes”, and

sometimes the “neck”). Thus the goal of active stereo head calibration is to obtain a mapping from 3D to the stereo images that is parameterized by the motion variables, typically a minimum of 3 or 4 degrees of freedom are used, (either pan and tilt for each eye [8], or, elevation, vergence, version [6]). The difficulty lies in the fact that while the controlled motion is known (to a certain degree of accuracy), the motion axes are typically not coincident with the axes of the camera/image coordinate system and thus any kinematic models must take this offset into account. The data collection and modeling steps must be extended to (1) incorporate the motion parameters into the camera model, and (2) collect data to enable determination of the model parameters.

In Li [9] the single camera calibration is performed at a number of camera positions (typically nine positions for each eye). The extrinsic parameters (rotation and translation of the camera with respect to the world) obtained from this calibration at *each* of the various pan/tilt positions are then used in a non-linear optimization performed to determine the transformation from the motion axes to the camera.

Thacker and Courtney[13] present a statistical framework to determine the camera model parameters. Much like a NN, the statistical framework allows continuous updates of the parameters.

Brooks *et al.* [2] describe a method to perform self-calibration of a robot head. They assume that the optical axes are co-planar and develop a model of the fundamental matrix in terms of the calibration parameters. Pan angles are *recovered*, (when, in practice the pan is under computer control). In a similar formulation, Knight & Reid [8] present a method to automatically align a stereo head. Point correspondences are used to find the projective mapping between the two images. This mapping is decomposed to determine the geometric parameters. Both of these methods requires a set of matching point correspondences between the two cameras, which, for wide baselines, can be difficult to obtain. Neither of these methods exploit the fact that the motion axes of the stereo head are typically under computer control and generally known with reasonably high accuracy.

2 Automated Data Collection

The equipment used to collect data consists of two Directed Perception pan tilt units with color Sony XC-999 CCD cameras and two Meteor frame-grabbers. The cameras and pan tilt units were mounted on an RWI-B21 mobile robot with about a 30cm baseline



Figure 1: The robot vision system

(see figure 1). A computer controlled LCD projector with 960 pixels by 720 pixels resolution was hung from the ceiling behind the robot. The distance between the projector and the wall was about 4 meters. The image of each pixel on the wall had a size about $4mm^2$.

The calibration patterns consisted of red and green circular blobs projected by a LCD projector at the appropriate positions. The red blob provides a fixation point for the cameras, while the green provides data off the optical axis. Then the cameras were fixated on the red blob and the pan and tilt angles of both PTUs along with the position of the centroids for the red and green blob in both images was recorded. The centroid of each blob can be estimated accurately up to sub-pixel accuracy [3].

Fixation consisted of moving pan-tilt units (PTUs) until the centroid of the red blob, or fixation point, was within one pixel of the center of both CCD images. After fixation was complete and data collected, the red and green blob were moved to a new position thus requiring a new fixation. In this way we ensured that the errors in the noise associated with measuring each set of project blobs was statistically independent.

At each of the robot's positions the same pattern of red and green blobs were presented to the robot. This pattern is refer to as a "sheet". Within each sheet, the red blobs were presented at 160 (10 vertical, 16 horizontal) regularly-spaced positions with 40 pixels increments in each direction.

The green blob position was defined relative to the red blob. For each red blob position on the sheet, the green blob had 12 possible positions: at ± 60 and ± 120 pixels in each the x or y directions relative to the red blob (8 of the possible positions). The other 4 were at each corner with the center at ± 60 pixel differences in both the x and y (see Figure 2(a)). A total of 1920 (160×12) different combinations of red/green

blob positions occur within each sheet and a sheet was presented every three centimeters over approximately a 15 centimeter total displacement of the robot.

The position of the robot was re-measured manually after every movement due to poor position control of the robot. The measurements were taken with two tape measures and two plumb weights. This allowed for accurate positioning of the robot and control of the rotation of the robot turret. The error in positioning of the blobs and robot was minimized as much as possible but could not be eliminated entirely. Obvious sources of errors include:

- Assumed Planarity: The blobs projected on a wall were assumed to lie on a plane. This was not the case due to small variations in the wall.
- Assumed fixed lateral robot position: It was assumed that the lateral position of the robot remained constant at the center of the projected image. Small measurement errors and possible misalignment of the plumb weights may cause errors in the robot's lateral position.
- Accurate wall to robot distance: The measured distance between the robot and the wall was affected by measurement error when reading from the tape measure.
- Angular resolution of PTU: Exact pan/tilt angles can not be determined but rather are measured to within the accuracy of the device.
- Image location of the blob centroids had small errors (although centroids can be detected to sub-pixel accuracy independent of thresholds chosen[3]).

3 NN for Visual Calibration

NN(neural networks) have previously been employed to improve the computation of 3D positions using stereo images [7, 4]. There are two general approaches. The first is a pure NN and the other is a hybrid system based on the combination of conventional calibration and NN. The total NN approach has the advantage of not having bias toward any particular camera model. The hybrid NN takes advantage of all the existing research with various camera models and uses the NN to improve these model by removing residue error. This paper will look use a pure NN approach.

Our NN was composed of three layers: input, hidden, and output. The input layer was composed of 8 neurons, four for each camera: ϕ , τ , u , and v . The

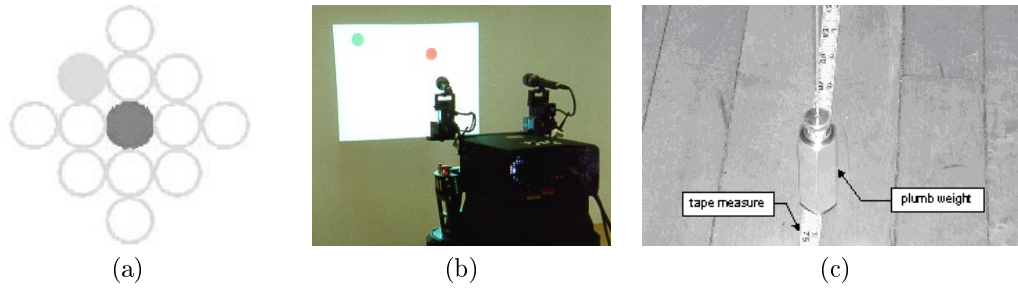


Figure 2: (a) Visual stimuli used for data collection. The center blob is the red blob. The fill dot represents an example position of the green blob and the rings represent the other possible positions of the green blob relative to the red blob. (b) The robot collecting data. (c) Plumb lines hung from the robot are used to manually collect robot position data to correct for inaccuracies in the robot odometry.

input layer was only partially connected to the hidden layer implying that each hidden neuron requires only a subset of the input layer’s output. The weight vector applied to the output of the input layer was not updated during the learning process, but rather remained constant.

The hidden layer was based Pouget’s [11] model of the visual processing neurons contained in the parietal cortex of the human brain. The activation function of each hidden neuron was modeled as a product of a sigmoid and a gaussian. This product is referred to as a basis function. There were two types of basis functions for each camera,

$$h_{u\phi}(i) = \frac{e^{-\frac{(u-u_i)^2}{2\sigma_u}}}{1 + e^{-\frac{\phi-\phi_i}{T}}} \text{ and } h_{v\tau}(i) = \frac{e^{-\frac{(v-v_i)^2}{2\sigma_v}}}{1 + e^{-\frac{\tau-\tau_i}{T}}}. \quad (1)$$

The first type of basis function was composed of a gaussian in u (the horizontal image pixel position), a sigmoid in ϕ (pan), and two thresholds u_i and ϕ_i . The second type of basis function was composed of a gaussian in v (the vertical image pixel position), a sigmoid in τ (tilt), and two thresholds v_i and τ_i . Each of the four types of threshold values were determined heuristically by selecting 11 uniformly spaced values over an interval slightly larger than the range of the corresponding input. Since there were two cameras each with two types of basis functions, and 11 different values for each of the four types of thresholds, the hidden layer was composed of $2 \times 2 \times 11 \times 11 = 484$ unique neurons.

The hidden layer was fully connected to the output layer. The output of the hidden layer (h_i), inputs to the the output layer, were multiplied by a weight vector. This weight vector was updated during the learning process using the standard back propagation. The output layer was composed of three neurons, each having a linear activation function. Each neuron cor-

responded to one of the output values x_w , y_w , or z_w of either the red or the green blob.

The value of σ_u was chosen such that if the value of u fell half way between two neighboring values u_i ’s, the sum of the values of the two gaussians would be very close to one. Thus the value of σ_u depended on the range of values and the number of u_i ’s, which in turn depends on the input data. The same holds true for the σ_v . The value of T for the basis functions was adapted thru trial and error using the T value of Guse [7] as a initial guess.

4 Active linear calibration of a pan-tilt camera

The transformations from world to image is modelled as a projective linear relationship. A world point, $(x, y, z, 1)$, projects to an image point (u, v) via the relationship

$$\begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = [INT][EXT] \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (2)$$

where $[INT]$ is a 3×4 projection matrix containing the intrinsic camera parameters, and $[EXT]$ is a 4×4 matrix containing the extrinsic camera parameters: i.e. the transformation from the world coordinate system to the camera coordinate system. We parameterize this transformation using three transformations (shown graphically in figure 3):

$$[EXT] = [T_{cam\ gaze} T_{gaze\ ptu} T_{ptu\ base}] \quad (3)$$

where $T_{cam\ gaze}$ is the transformation from the pan-tilt units to the camera frame (a fixed transformation determined via calibration with Li’s method),

$T_{gaze\ ptu}$ represents a pure rotation of the PTUs (this is parameterized by the pan and tilt angles), and $T_{ptu\ base}$ is the mapping from the home position of the pan-tilt units, ptu , to the base coordinates. The system formulation is considerably simpler if the camera can be made to rotate about its optical center, however, such precision in placement of a camera is improbable (and is not the case with our hardware).

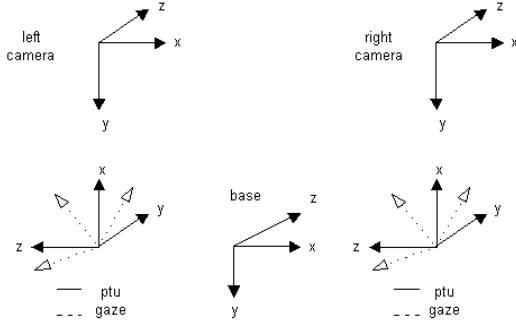


Figure 3: The coordinate frames used for the stereo head. Note that the motion of the PTUs is not coincident with the optical center of the imaging equipment.

Thus, given the pan, tilt angles for the left camera, (ϕ^L, τ^L) , we solve for $[EXT]$. Then using the measured image points (u^L, v^L) we substitute the measured/controlled values into eq. 2 and eliminate the λ to obtain two equations for the three unknowns $[x, y, z]$. Repeating for the right camera also yields two equations. The four equations for the three unknowns $[x, y, z]$ were solved using total least squares. The linear method is conceptually simple, however in order to solve for $[EXT]$, the transformation $T_{cam\ gaze}$ must be obtained. One calibration procedure [9] that solves for this transformation uses a “classical” calibration procedure [14] (observing a calibration pattern and recovering the 2D-3D correspondences) at many (~ 10) different pan-tilt positions to obtain the $T_{cam\ world}$ for each camera position. By knowing the motion of the PTUs relative to some initial home position and calibrating the extrinsic parameters at each of the controlled positions, a non-linear optimization procedure was used to extract the transformation from the home position to the camera coordinate frame.

5 Experimental Results

In this experiment we deliberately used a large number of data samples so that effectiveness of the proposed methods can be analyzed while minimizing the affect of statistical sampling error.

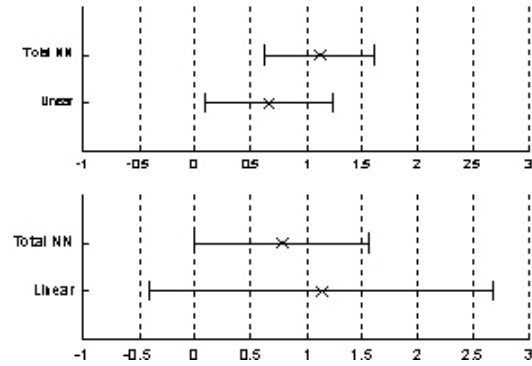


Figure 4: The table plots the mean error and standard deviation in x_w (in centimeters). The top chart contains the data for the red blobs and the bottom chart for the green. The average absolute error is indicate by \times and the length of the line is two standard deviations (i.e. we show $x_w \pm \sigma$).

Five sheets of data samples were collected. The red blob data (samples on the center ray) were processed separately from green blob (sample off the center ray). The data from each sheet was put into one of two categories. The data from first ($z_w = 162.20cm$), third ($z_w = 168.15cm$), and fifth ($z_w = 174.00cm$) sheets were used as data for training the NN. The data from the second ($z_w = 165.10cm$), and fourth ($z_w = 171.00cm$) sheets were used for testing. The testing sets were not used in the training of the NN in a effort to prevent memorization of the examples or over-training.

Figures 4, 5, 6, and table 1 present errors in 3D reconstruction using the reconstruction method of section 4 and the NN method described in section 3. This table also presents the standard deviation, σ , of the data, which is useful in characterizing the spread of the data. For the x_w and y_w directions we compare the calculated coordinates with the coordinates measured on the wall (assumed to be the “true” coordinates). For the x_w and y_w directions the error is computed as follows. Let \bar{p}_i be the vector average of all the calculated coordinates $p_j = (x_j, y_j)$ for position i , where $i = 1, \dots, 160$, indexes one of the 160 unique red positions within each sheet, i.e. $\bar{p}_i = \sum_{j=1}^{12} (x_j, y_j)$. Then the mean error is

$$\mu = \frac{1}{160} \sum_{i=1}^{160} |\bar{p}_i - p_{measured}|$$

where $p_{measured}$ are the “true” coordinates of the blobs centroid as measured on the wall. Similarly the total standard deviation is computed using the aver-

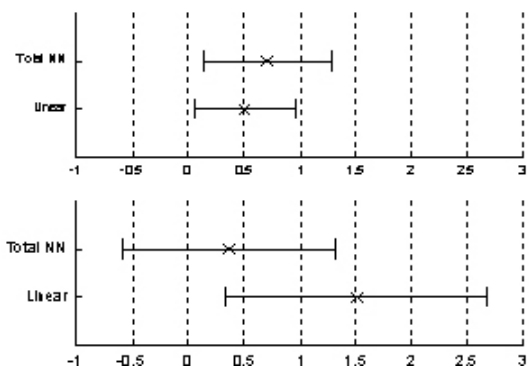


Figure 5: Using the same conventions stated in figure 4 the error and standard deviation in the y_w direction.

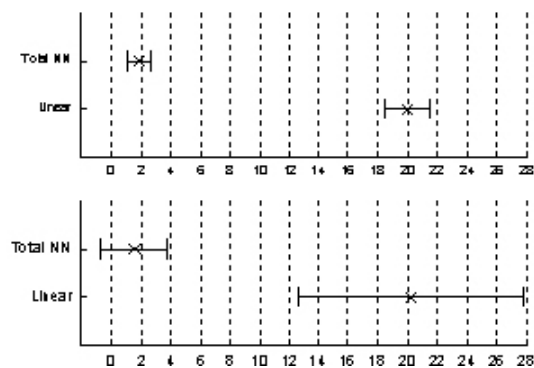


Figure 6: Error and standard deviation in the z_w direction.

age of the standard deviations of each of the 160 positions. Because the “true” depth is constant within each sheet, we compute the error statistics in the z_w direction for the red blob by averaging z_w coordinates of all blobs within a sheet. Thus we compute the z_w error using

$$|\bar{z}_w - z_{measured}| \quad \text{where} \quad \bar{z}_w = \frac{1}{1920} \sum_{i=1}^{1920} z_{w_i}$$

is the average calculated z_w for all the red blobs in a sheet. Similarly the standard deviation is computed for the the calculated z_w over one entire sheet.

The error in the x_w and y_w of the green blobs was found relative to the red blobs projected with them; thus there are 12 unique positions for the green blob. The formulas above are applied to the *relative* green position data and for green blobs, \bar{p}_i is the difference between the calculated position in the world coordinate system of the red blob and the green blob at position i , where i is one of the 12 unique positions. The error and σ for z_w was found in the same manner as the red blobs.

From table 1 it can be seen that the average error and σ in the green blobs was larger than that in the red. Two factors contribute to this larger error: 1) the green blob’s x and y position is described with respect to the calculated red blobs position thus error in the red blob’s position increase the green blob’s error, and 2) the green blob’s reconstruction requires intrinsic parameters such as focal length, which were not used for reconstructing the red blobs coordinates because they were always on the optical axis.

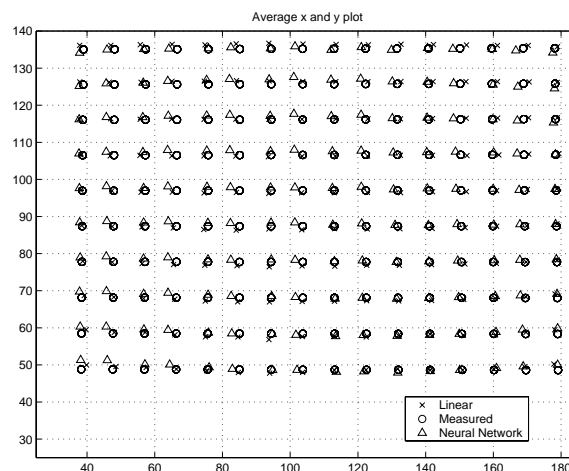


Figure 7: The x_w and y_w position of the red data is shown for the linear method, NN, and measured positions in cm.

6 Discussion

The NN was trained with the data gathered using an automated data collection procedure which provided a easy and quick (approximately 4000 samples per hour) method to gather large amounts of data. The NN requires a large amount of computer time (6 hours total for training and data acquisition), but very little operator time.

The linear calibration method requires several images of a non-coplanar calibration pattern to be obtained with differing pan and tilt angles for each camera. The features in the image then need to be matched with the known world coordinates to generate a minimal set of data to calibrate the cameras. To ensure that no correspondence errors occur, manual intervention is required for each image, requiring large amounts oper-

Method	Blob	Error	x_w	y_w	z_w
Linear	red	σ	0.57	0.45	1.45
		μ	0.67	0.50	19.9
	green	σ	1.54	1.71	7.61
		μ	1.14	1.51	20.22
NN	red	σ	0.50	0.57	0.82
		μ	1.12	0.71	1.92
	green	σ	0.79	0.96	2.24
		μ	0.78	0.37	1.60

Table 1: Comparison of accuracy for each coordinate axis and blob color (fixated *vs* non-fixated targets). The mean error, μ and standard deviation, σ , in the reconstructed value compared to the actual 3D coordinates are given (cm).

ator time. This method does not require any training and thus used less computer time than the NN.

To compare the linear-stereo and NN methods of reconstructing 3D points from active stereo images we used the “extra” data gathered, (the sheets that were not used in the training of the NN). The performance of both methods was similar in the x_w and y_w direction. The mean error and standard deviation in the x_w and y_w directions was on the order of one centimeter for both methods, which is reasonable for points 1.60 - 1.75 meters from the camera. However, in our experiments, the depth computation results using the linear-stereo method contained a unreasonable amount of error, generally an order of magnitude larger than using the NN. The NN yielded reasonable results in depth, z_w , with average errors of about two centimeters (about 1-2 % of the object distance). Using a neural network (NN) may be an attractive alternative to explicit camera models when reconstruction 3D using an active stereo head.

References

- [1] F. Angrilli, S. Bastianello, and R. DaForno. Calibration of stereo vision systems by neural networks. In *IMTC Conf.*, 839–842, 1996.
- [2] M.J. Brooks, L. de Agapito, D.Q. Hyunh, and L. Baumela. Towards robust metric reconstruction via a dynamic uncalibrated stereo head. *J. Image and Vision Computing*, 16:989–1002, 1998.
- [3] P. I. Corke. *Visual Control of Robots*. Research Studies Press Ltd., 1996.
- [4] Y. Do. Application of neural networks for stereo-camera calibration. In *Int'l Joint Conf. on Neural Networks*, 2719–2722, 1999.
- [5] O. D. Faugeras and G. Toscani. The calibration problem for stereoscopic vision. In *Sensor Devices and Systems for Robotics*, volume 52 of *NATO ASI*. Springer-Verlag, 1989.
- [6] N.J. Ferrier and J.J. Clark. The harvard binocular head. *Int'l Journal of Pattern Recognition and AI*, 9–32, March 1993.
- [7] N. Guse. A neural network for visual kinematics. MS Thesis, Univ. of Wisconsin-Madison, 1999.
- [8] J. Knight and I. Reid. Active visual alignment of a mobile stereo camera platform. In *IEEE Int'l Conf. on Robotics and Automation*, 3203–3208, San Francisco, CA, April 2000.
- [9] M. Li. Kinematic calibration of an active head-eye system. *IEEE Trans. on Robotics and Automation*, 14(1):153–157, February 1998.
- [10] H.A. Martins, J.R. Birk, and R.B. Kelley. Camera models based on data from two calibration planes. *Computer Graphics and Image Processing*, 17:173–180, 1981.
- [11] A. Pouget. *Computational Models of Spatial Representations*. PhD thesis, University of California, San Diego, 1994.
- [12] S.-W. Shih, Y.-P. Hung, and W.-S. Lin. When should consider lens distortion in camera calibration. *Pattern Recognition*, 28(3):447–461, 1995.
- [13] N.A. Thacker and P. Courtney. Online calibration of a 4 dof stereo head. In *British Machine Vision Conference*. Springer-Verlag, 1992.
- [14] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Trans. of Robotics and Automation*, RA-3(4):323–344, August 1987.
- [15] J. Weng, P. Cohen, and M. Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(10):965–980, 1992.
- [16] Y. Yakimovsky and R. Cunningham. A system for extracting three-dimensional measurements from a stereo pair of TV cameras. *Computer Graphics and Image Processing*, 7:195–210, 1978.