ELSEVIER

Image and Vision Computing 20 (2002) 619–630

image AND vision COMPUTING

www.elsevier.com/locate/imavis

A neural approach to zoom-lens camera calibration from data with outliers

Moumen Ahmed, Aly Farag*

CVIP Laboratory, University of Louisville, Louisville, KY 40292, USA

Received 10 June 2001; received in revised form 14 March 2002; accepted 14 March 2002

Abstract

Camera systems with zoom lenses are inherently more useful than those with passive lenses due to their flexibility and controllability. However, their calibration raises several challenges. In this paper, we present a neural framework for zoom-lens camera calibration that can capture complex variations in the camera model parameters across continuous ranges in the lens control space, while minimizing the calibration error over all the calibration data. To automate the tedious process of collecting calibration data, the calibration approach should be prepared to handle possible outliers in the data. We demonstrate how the calibration approach can be robust and less sensitive to outliers. The validity and performance of our approach are tested using both synthetic data with outliers, and with real experiments to calibrate Hitachi CCD cameras with H10 × 11E Fujinon active lenses. © 2002 Published by Elsevier Science B.V.

Keywords: Camera calibration; Zoom-lens; Active vision; Robust statistics

1. Introduction

One goal of machine vision is to understand the visible world by inferring 3D properties from 2D images. Making such an inference requires modeling of the relationship between the 2D images and the 3D world. Camera calibration is a process which models this relationship. In this work, we report our approach to calibrate zoom-lens cameras that are part of the CardEye, an active vision system developed at our lab, see Fig. 1. The system has a trinocular head with basic mechanical properties such as pan, tilt, roll, focus, zoom, aperture, vergence and baseline. The system uses an active lighting device (a laser pattern generator with different diffractor filters), located at the center of the head, to assist in surface reconstruction process and other tasks. The system makes use of the flexibility and controllability of the zoom-lens cameras to improve the performance of the system in its tasks. To effectively use active vision systems such as ours, we need to be able to build accurate camera models that hold calibration across continuous ranges of one, two, or even three lens parameters. However, calibrating cameras with zoom lenses is rather difficult and raises several challenges. First, the dimensionality of data collected for calibration is large. Second is the potential difficulty in taking measurements across a wide range of imaging conditions (e.g. defocus and magnification changes) that can occur over the range of zoom and focus control parameters.

The calibration problem of these cameras relies on formulating functions that describe the relationships between the camera model parameters and the lens settings. This is usually achieved by calibrating a conventional static camera model (commonly the pinhole model) at a number of lens settings which span the lens control space using traditional calibration techniques. The calibrated model parameters at each lens setting are then stored in lookup tables [2,8], or polynomials (or perhaps other functions) are formulated to model the parameters [4,5,9]. The following remarks can be drawn on the previous approaches:

- Using interpolation to obtain each model parameter at intermediate lens settings in tables or fitting a function to each parameter independently emphasizes only the fitting error for this particular parameter at the expense of the overall calibration accuracy, which is completely ignored in this step. Subsequently, this methodology does not consider the interaction between all the model parameters to represent the underlying camera model.
- Polynomials in many cases fail to follow the complex variations in some model parameters [2,10]. Although other alternatives such as exponential functions, Chebyshev polynomials and Legendre polynomials can be exploited, the question about the optimal (or best)

* Corresponding author.
E-mail addresses: farag@cvip.uofl.edu (A. Farag), moumen@cvip.uofl.edu (M. Ahmed).

0262-8856/02/$ - see front matter © 2002 Published by Elsevier Science B.V.
PII: S0262-8856(02)00051-3

Fig. 1. CardEye trinocular active vision system: From left to right, the trinocular head and the control circuitry cabinet.

function type often remains difficult to answer. Look-up tables have been previously used to get around this problem.

To remedy the first point, an additional step of global optimization over all the calibration data is carried out in Refs. [4,9,10] to refine the coefficients of fitted polynomials. To avoid the problem of optimizing a large number of coefficients of all polynomials at the same time, this step optimizes each polynomial in turn until one (or more) cycle is completed for all the parameters. However, as noted in Ref. [4], the sequence of fitting the polynomials to the data affects the final calibration error. Therefore, several experiments may be needed to reach a good sequence.

No a priori knowledge about the shape of the various camera model functions can be assumed available since such information is seldom revealed by manufacturers. Furthermore, in mass production, optical characteristics are sure to vary from one lens to another. Therefore, the actual relationships between the camera model and the control settings must be determined empirically through taking several calibration measurements (world to point correspondences) over working ranges of lens settings. In this work, we present a flexible method to represent any model function on continuous ranges of lens control space. In particular, we resort to the proven power of multi-layered feedforward neural networks (MLFNs) as universal approximators [6] to provide suitable parameter formulation/fitting. This can take care of the second point, but one still needs to consider the interaction between these MLFNs in such way that minimizes the calibration error over all data. We have recently [11] devised a novel solution by introducing the neurocalibration approach, which casts the camera calibration problem into a learning problem of an MLFN. Therefore, we propose to add the neurocalibration network to a number of MLFNs, thus building an all-neural framework in which all the MLFNs learn concurrently to capture the variations of model parameters across continuous ranges of lens settings. This framework can provide handy solutions to the previous two points. We demonstrate its performance on the CardEye's Hitachi cameras with Fujinon lenses.

In an addition to a brief description of our approach to zoom-lens calibration, in this paper, we address another related problem. Collecting calibration data that covers the working space of the system is rather tedious. Therefore, we are planning to have the system collect the data by itself. This can be accomplished by using special markers embedded in the system working space. Moreover, we are currently investigating the integration of the active lighting device with a range finder such that the two use the same laser beam. The integrated active device will have two modes: as pattern generator and as range finder. Such a device is useful in both surface reconstruction and obtaining data for calibration. Automating the data collection phase makes the system easy to re-calibrate whenever necessary. However, it may introduce outliers in the data due to poor localization of the markers or the laser pattern in the captured images, or due to incorrect correspondence automatically established between the 3D points and their corresponding image pixels. It is worth mentioning here that usually off-line calibration is done under human supervision and outliers are less likely to occur. That is why, to the best of our knowledge, we are not aware of any previous calibration work that has dealt with outliers in the data.

To cope with the new possibility of outlying calibration data, a robust approach to calibration should be pursued; one bad outlier would skew the results of any approach based on the widely used least squares estimates. We therefore propose to use the maximum likelihood estimator *M-estimator* [13] as a more robust estimator.

The rest of this paper is organized as follows. We describe the camera model and state the calibration problem in Section 2. In Section 3, we briefly describe the neurocalibration network, followed by a more robust approach in Section 4. A brief overview of the framework for zoom-lens calibration is given in Section 5. Sections 6 and 7 describe our experiments and our concluding remarks.

2. Camera calibration problem

The result of camera calibration is an explicit transformation that maps a 3D world point $\mathbf{M} = (X, Y, Z, 1)^{\mathrm{T}}$ into a 2D pixel $\mathbf{m} = (u, v, 1)^{\mathrm{T}}$. This mapping can be represented by a 3×4 projection matrix $\mathbf{P}$ that encompasses 11 physical parameters: rotation angles R_x, R_y and R_z, translations t_x, t_y and t_z, the coordinates of the principal point (u_0, v_0), two scale factors α_u and α_v, and the skew c between the image axes. This camera model thus ignores lens distortion which is often accounted for in the camera model by adding some distortion parameters [1]. However, with the state of the art of the technology, camera distortion is reasonably small, and the pinhole model is thus a good approximation. On the other hand, if lens distortion is noticeable (this may be the

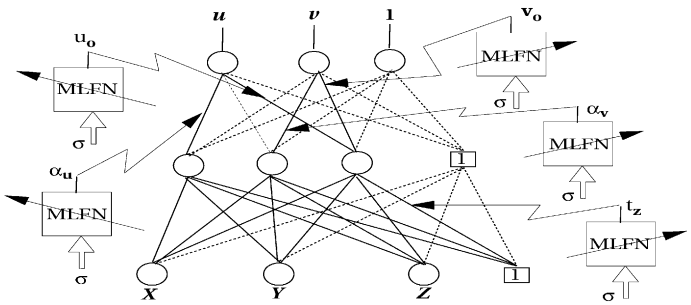

Fig. 2. Global optimization step: five parameter MLFNs cooperate with the central neurocalibration net during the optimization, σ denotes the lens controllable optical parameters and dashed links in the central network designate link weights fixed during learning at 0 or 1.

case at small camera focal length), the distortion parameters can be estimated in the captured images by a pre-calibration process [3,7] keeping in mind that these parameters may vary with lens settings. Then the images or image features can be undistorted before calibration proceeds. In this work, we follow this strategy for a couple of reasons. First, the decoupling between distortion parameters from the other model parameters will allow us to maintain the simple relationship of the distortion-free model thus making subsequent vision tasks (e.g. stereo reconstruction) easier. Moreover, the decoupling would reduce the effect of the correlation between lens distortion coefficients and other camera model parameters [10] on parameter estimation.

Given a sufficient number, N, of reference world points, $\mathbf{M}_i = (X_i\ Y_i\ Z_i\ 1)^{\mathrm{T}}$, as well as their corresponding pixel positions, $\mathbf{m}_i = (u_i\ v_i\ 1)^{\mathrm{T}}$, the camera calibration problem is to estimate the 11 camera parameters or the projection matrix $\mathbf{P}$, that minimize

$$E^{\text{calib}} = \sum_{i=1}^{N} \left(\frac{\mathbf{P}^1\mathbf{M}_i}{\mathbf{P}^3\mathbf{M}_i} - u_i \right)^2 + \left(\frac{\mathbf{P}^2\mathbf{M}_i}{\mathbf{P}^3\mathbf{M}_i} - v_i \right)^2, \tag{1}$$

where $\mathbf{P}^i$ denotes the ith row of $\mathbf{P}$.

In Section 3, we give a summary of an MLFN that solves this problem. However, since the camera calibration parameters may vary as the lens setting is changed, the calibration problem of a zoom-lens camera system becomes finding the intrinsic and extrinsic camera parameters, expressed as functions of the controllable camera settings, which can be composed for any fixed camera setting in order to obtain the projection matrix. This problem is addressed in Section 5.

3. Neurocalibration

In Ref. [11] we proposed an MLFN that not only learns the perspective projection mapping of a camera, but also can solve explicitly for the calibration parameters. Here, we give a brief summary of our approach. The interested reader can refer to [11] for more details. The neurocalibration net has a topology of 4–4–3 with linear hidden and output neurons (see the central net in Fig. 2). The weight matrix of the hidden layer is denoted by $\mathbf{V}$, and it is assumed to correspond to the extrinsic parameters. The weight matrix of the output layer is denoted $\mathbf{W}$ and it corresponds to the intrinsic parameters. For any input pattern $\mathbf{M}_i$, the network output, (o_{ik}, $k = 1,2,3$), represents the 2D pixel homogeneous coordinates. With a literal interpretation of the error criterion in Eq. (1), the error measure here would be

$$E^{\text{calib}} = \sum_{i=1}^{N} \left(\frac{o_{i1}}{o_{i3}} - u_i \right)^2 + \left(\frac{o_{i2}}{o_{i3}} - v_i \right)^2. \tag{2}$$

This last equation is not in a form that can be minimized easily by an MFNN due to the presence of the two ratios in terms of the network outputs. To approach this problem, we introduce a new variable γ_i for each point and train the network such that $\gamma_i o_{i3}$ approaches 1 for each point. This is done by taking the network error criterion as

$$E = \sum_{i=1}^{N} (\gamma_i o_{i1} - u_i)^2 + (\gamma_i o_{i2} - v_i)^2 + (\gamma_i o_{i3} - 1)^2. \tag{3}$$

One can look at γ_i as the slope of the linear activation function of the output neurons, which is generally different from one input point to another. The network should solve for not only the values of the weights, but also values of γ_i, for all i, that minimize Eq. (3). As network training proceeds, $\gamma_i o_{i3}$ will approach 1 and the error in Eq. (3) will thus be eventually equivalent to the geometric (Euclidean) calibration error E^{calib} in Eq. (1).

Let $\mathbf{d}_i = (u_i\ v_i\ 1)^{\mathrm{T}}$ and $\mathbf{O}_i = \gamma_i \mathbf{o}_i$. Then the error measure can be put in the familiar form

$$E = \sum_{i=1}^{N} \sum_{j=1}^{3} (d_{ij} - O_{ij})^2. \tag{4}$$

The weights of the network W_{kj} and V_{lj} are initialized at random values in the range $-1{:}1$, while all the different γ_i are initially set to 1. The network weights W_{kj}, V_{lj} and γ_i are updated according to the gradient descent rule applied to Eq. (3) [11]. For ease of network learning, the input and desired patterns of the network are normalized by s_1 and s_2, respectively. After training the network, the projection matrix $\mathbf{P}$ can be shown to be [11]

$$\mathbf{P} = \mathbf{S}_1\mathbf{W}\mathbf{V}\mathbf{S}_2, \tag{5}$$

where

$$\mathbf{S}_1 = \mathrm{diag}(s_1, s_1, 1), \quad \text{and} \quad \mathbf{S}_2 = \mathrm{diag}(s_2^{-1}, s_2^{-1}, s_2^{-1}, 1). \tag{6}$$

Moreover, in order to have the network explicitly solve for the camera model parameters (not just matrix **P**), each network weight is mapped to one camera parameter by enforcing the orthogonality constraints on **R** during network learning. The constraints are represented as additional terms added to the error criterion to be minimized. The new error measure will be

$$E_{\mathrm{tot}} = E_{\mathrm{2D}} + \beta E_{\mathrm{orth}}, \tag{7}$$

where E_{2D} is the same in Eq. (3) and E_{orth} is a sum of six error terms [11] that ensure the correct submatrix of **V** to be a rotation matrix. As such, the matrix **V** will indeed include the actual camera extrinsic parameters. The positive weighting factor, β, increases slowly as learning proceeds.

The network is trained by the traditional backpropagation algorithm, however, speedup can be achieved by applying the conjugate gradient method during some periods of the training process. Switching between conjugate gradient and gradient descent can be done automatically.

Our extensive simulations and tests on practical images [11] yielded very low calibration error and have shown that this neurocalibration approach has the following features:

- It relaxes the requirement of a good initial starting point, which is common to other non-linear optimization techniques (e.g. Levenberg–Marquardt algorithm). These techniques often fail without this condition. In all the experiments conducted, the network has converged starting from random initial weights without sacrificing the calibration accuracy.
- Experiments have shown very small sensitivity of the network learning to network parameters, e.g. learning constants.
- The optimization procedure takes account of the structure of the orthonormal rotation matrix without extra optimization steps (e.g. the one used in Ref. [1]).
- It is simple; the reader can easily reproduce our code.
- The technique is completely parallel, thanks to its neural basis.

4. Robust neurocalibration

Various machine vision algorithms found in literature optimize a least squares criterion, which is optimum and reliable when the underlying noise in the data is Gaussian. However, when outliers are present in the data, the Gaussian assumption is violated and the least squares result is skewed. During the last three decades, many robust techniques have been proposed [12–14] to handle outliers and these techniques have gained popularity in computer vision [18]. Robust estimates include M-estimates (maximum likelihood estimates), L-estimates (linear combination of order statistics), R-estimates (estimates based on rank transformations) and LMedS estimates (least median square). If r_i denotes the residual error of the ith data item, M-estimators try to reduce the effect of outliers by minimizing another function of the residuals, $\sum_i \rho(r_i)$, where ρ is a symmetric, positive-definite function with a unique minimum at zero, and is chosen to be less increasing than square. Many of such ρ function have been suggested [13,14], which yield breakdown points of about $1/p$, where p is the number of unknowns ($p = 11$ in case of camera calibration). M-estimators have high efficiencies, typically 0.9 [14] (efficiency is defined as the ratio between the lowest achievable variance for the estimated parameters and the actual variance provided by the given method [14]).

To robustify our calibration approach, instead of minimizing the error in Eq. (4), the network will minimize

$$E = \sum_{i=1}^{N} \sum_{j=1}^{3} \rho(d_{ij} - O_{ij}). \tag{8}$$

We selected the redescending function suggested by Tukey [14] for the function ρ, which has provided better results as compared to other function such as Huber's function [13]. The updating rules for network weights will be different from those used in Ref. [11] and are re-derived according to the new error in Eq. (8), see Ref. [17] for more details.

It is important to note that M-estimate methods tend to be extremely susceptible to the initial solution to the non-linear optimization method. Most calibration approaches found in the literature initialize a non-linear optimization algorithm with the closed-form solution of a least squares linear approach. However, due to outliers, the linear approach would not provide a reliable initial solution to the non-linear optimization algorithm, which subsequently fails to yield any improvement. Our neural approach does not suffer from this problem since it starts with random initial weights.

After network training, one can make a good, *robust* estimate of the standard deviation of the errors of good data (inliers). This estimate is related to the median of the absolute values of the residuals, $\hat{\sigma}$ [12]. Any data item whose error is larger than a certain number (e.g. 2.5–3.0) of $\hat{\sigma}$ can be considered as an outlier and removed. We are currently investigating another approach based on LMedS estimates [12], which theoretically has the largest possible breakdown point (0.5) but lower efficiency and higher computational complexity.

5. Zoom-lens camera calibration

One can precisely state the calibration problem for a zoom-lens camera as the problem of finding the intrinsic and extrinsic camera parameters, expressed as functions of the

adjustable camera settings such as zoom, focus and/or aperture, which can be composed for any fixed camera setting to obtain the projection matrix **P**. In this way, the elements of the projection matrix will be non-linear functions of the explicit physical camera parameters, which are functions of the camera settings over the camera setting range.

One way to solve this problem [2,8,10] is to store the calibrated camera model parameters at each lens setting in a look-up tables. As such, one need not worry about finding formulas for how they vary with the lens setting. However, one would readily see at least two drawbacks of this strategy. First, without an algebraic form for each model parameter, it would be rather difficult to obtain values for any parameter across continuous ranges of the control space. Although interpolation can be used to solve for the parameter value given its values already stored in the table, this would ignore the interaction between all model parameters at any given lens setting to form the perspective mapping of the camera. This observation highlights the second drawback, which can be evaded by accounting for the parameter cooperation and interaction in deriving the algebraic formulas of the parameters. The decisions on the proper formulas to use may be based on design objectives, such as requiring a particular parameter to be constant for all lens setting. However, in many cases these decisions must be made empirically by examining the data [4].

One can regard zoom-lens camera calibration as a combination of fixed-parameter camera calibration and function interpolation over a large collection of data that cover the camera setting ranges. We resort to the known fact that MLFNs are universal approximators [6] to provide suitable and flexible parameter formulation. Each MLFN, called *parameter* MLFN from now on, will provide the functional relationship between one camera parameter and lens settings. To link the different parameter formulas and to include the coupling between the camera parameters, we use the neurocalibration network in an explicit calibration mode in which each parameter is mapped to a network weight. If a model parameter is to be modeled with a constant over all lens settings, it will be represented in the neurocalibration network by the corresponding network weight as usual. Whereas for any parameter that is allowed to vary as a function of lens setting, the corresponding network weight is replaced by the output of the corresponding parameter MLFN. Whenever necessary, the parameter MLFNs' outputs will act as regular neurocalibration network weights, after proper scaling if necessary, in order to compute the output and calibration error. The non-typical resultant neural network structure is used to refine the mapping captured by the parameter MLFNs and to optimize the different parameter formulations in terms of the calibration error over all data.

In this section, we outline our framework for zoom lens calibration in the following three steps.

5.1. Data collection and passive camera calibration

The calibration process starts with collecting the calibration data at a number of different lens optical settings (zoom, focus and/or aperture) covering the operating space of the system. Since a priori knowledge about the shape of the various camera model functions is generally not available it it better to have as many samples of the control space as possible to improve the accuracy of the adjustable camera model. At each fixed lens setting, the fixed camera model parameters are estimated using the neurocalibration technique (any other calibration technique may be used for this step). If data is likely to have outliers, the robust neurocalibration technique is used, then outliers are identified and thrown away.

5.2. Initial parameter formulation

Having obtained the parameter values at the different positions, we are ready to fit functions to these values. The skew, c, is usually very close to zero in practice and it is thus fixed to zero in the camera model. For ease of use of the camera model, excluding t_z, the position and the orientation of the camera coordinate frame relative to the world coordinates are kept unchanged as the lens parameters are varied [4,10]. Therefore, R_x, R_y, R_z, t_x and t_y are modeled with constants (zero-order terms). Constraining these parameters to be independent of optical settings makes use of the extra-degrees of freedom in the calibration (i.e. the dependency and correlation between some camera parameters for small variations [10]). The initial value for any of the former five terms, say R_x, is set to the value $\bar{R}_x$ that minimizes $\sum_i (\bar{R}_x - R_x^{(i)})^2$, where i indexes each sampling position of the control space. Obviously, this is the mean value of each parameter throughout the whole data. Then, for each of the remaining parameters[1], α_u, α_v, u_0, v_0, and t_z, a function is fitted across the optical settings using an MLFN. Here we make use of the fact that a two-layer (one hidden layer) feedforward neural network is able to approximate any real-valued continuous function over a compact (bounded and closed) set up to any accuracy [6]. Each parameter MLFN has one output unit and in addition to the fixed bias, it has as many input units as the number of lens control parameters. Although a two-layer network is theoretically able to approximate any surface, it may require an impractically large number of hidden units in the case of complex surfaces [15]. It has been found, however, that an adequate solution can be obtained with a tractable network size by using more than one hidden layer [15]. Thus, the number of layers (one or more) and number of hidden units per layer are determined experimentally for each MLFN alone. Note that these parameter MLFNs, unlike the

[1] If appropriate, some of them may be modeled with constants as well. However we will follow this assumption because this is what we have found to suit the cameras we experimented with.

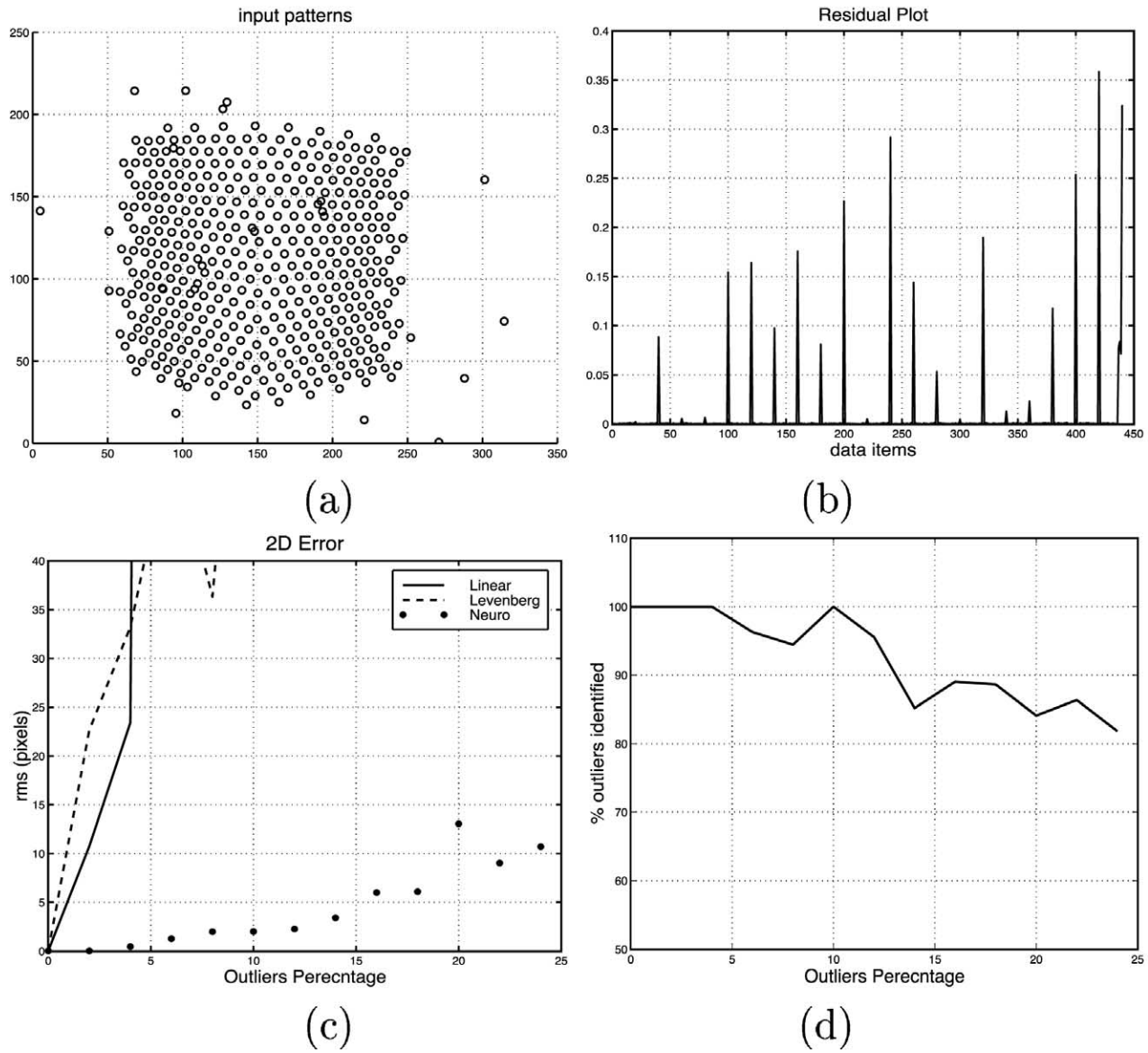

Fig. 3. Robust calibration: (a) 2D points with 5% outliers, (b) Residual plot of error at $\eta = 5$, (c) rms of point errors in pixels for three approaches and (d) Outlier identification rate versus η.

neurocalibration network, have non-linear activation functions. Each MLFN is trained independently without the necessity of attaining a very low fitting error since this training is just an initial step before the global optimization step.

5.3. Global optimization

In the previous step, each model parameter has been characterized by a suitable function independently from the other parameters. That step, accordingly, ignores the cooperation among these parameters to perform the perspective projection mapping of the camera at every lens setting. The global optimization step fixes this by tuning these functions such that the calibration error over all lens settings is minimized.

The neurocalibration net has a central role in the global optimization step. The five zero-order terms are represented by five network weights, while the five parameter MLFNs serve to provide the central network with the rest of the weights representing the parameters that vary with the lens optical setting. Fig. 2 illustrates the central neurocalibration network and its associated MLFNs.

The novel network of neural networks of Fig. 2 presents a non-typical structure of feedforward neural networks. We have developed an extended variant of the known back-propagation algorithm to train these networks minimizing the overall calibration error for all the collected calibration data.

At each lens setting, σ_i, the outputs of the five parameter MLFNs provide the corresponding five model parameter values to the central network, which uses these values along

with its weights representing the zero-order term parameters to compute the calibration error at this lens setting and then to update all its weights according to the gradient descent rule. Since five of those weights come from the outputs of the parameter MLFNs, each of which is considered as the new desired value of an MLFN output and is used to update the weights of the network and thus its functional mapping between the parameters and the lens settings. Therefore, the formulations of the parameters are updated in this step in a way that minimizes the overall calibration error (this may result in an interesting situation inside the MLFNs, which is equivalent to an automatic change of the type and order of the basis functions used for parameter formulation from the initial ones obtained in the previous step). Note that each parameter MLFN minimizes its own fitting error, which is different from the calibration error computed by the central neurocalibration network. However, the fitting error of each parameter MLFN affects the calibration error. Algorithm 1 shows an outline of the global optimization step.

Algorithm 1
Outline of global optimization step.

repeat
  $CycleError = 0$, $Ferror_k = 0$, $1 \le k \le 5$
  for all lens settings σ_i **do**
    The outputs of the parameter MLFNs are taken as 5 weights of the neurocalibration network.
    $SettingError = 0$.
    for all input–output patterns at σ_i **do**
      Compute calibration error, ε, from Eq. (7).
      All weights of central network are updated according to the neurocalibration updating rules [11].
      $SettingError += \varepsilon$.
    end for
    $CycleError += SettingError$.
    The updated 5 network weights are taken as new desired values and propagated back to the corresponding MLFNs.
    Each parameter MLFN k updates its own fitting error $Ferror_k$ using the difference between the new desired value and its output, and performs a learning iteration updating all its weights according to the standard backpropagation algorithm.
  end for
until no further improvement can be achieved in $CycleError$ and $Ferror_k\}$, $\forall k$.

While it is rather difficult to prove mathematically, the training process of all networks has indeed converged to a small calibration error in all of our experiments. Each parameter MLFN, then, will have the final functional relationship of that particular parameter versus lens settings while the central network will have the final values of the zero-order term parameters.

6. Experimental results

In this section, the performance of the robust calibration technique is tested with synthetic data with outliers. Then real experiments to calibrate the CardEye's cameras and to validate the calibration results are briefly described.

6.1. Calibration from synthetic data with outliers

Using a set of specific external and internal camera parameters, a set of 440 3D points are projected to form a set of 2D points in an image of size 320×243. Then, Gaussian noise with standard deviation 0.2 pixels is added to the 2D image coordinates to simulate the uncertainty in detecting these 2D points. A fraction, η, of the 2D points is selected and replaced with points generated randomly from uniform distribution that spans the whole image. Using the set of 2D points with the introduced outliers and the set of 3D points, we have performed a series of calibration experiments using different η. Fig. 3(a) shows the set of 2D points at $\eta = 5$ and the obtained residual plot of the points after minimization of the error in (Eq. (8)) using our network is depicted in Fig. 3(b). The points with gross error correspond to the actual outliers in the data, and thus they can be identified and removed. For comparison sake, two other calibration techniques have been tested: a linear calibration approach and a non-linear technique that starts with the solution of the linear approach minimizing the error in Eq. (1) using the well-known Levenberg–Marquardt algorithm. For each approach, the obtained projection matrix is used to project the 3D points. Then the *rms* error of the deviations between the point projections and the correct ones is computed and plotted against η in Fig. 3(c). Both the linear and Levenberg–Marquardt algorithm minimize a least squares criterion and thus they are largely affected by outliers in the data. The robust neurocalibration approach is less sensitive to outliers and produces error within 2 pixels until about 12% outlier percentage, which is around the theoretical upper bound of the breakpoint (9%). Fig. 3(d) shows the performance of our approach in terms of the percentage of correctly identified outliers versus η. Using an approach based on the LMedS estimator or the RANSAC framework is expected to improve further the breakdown point, but at the expense of increased computational complexity. Utilizing one of those more robust approaches depends to a large extent on the expected proportion of outliers in the input data, which still needs more investigation once the data collection mechanism is installed in our system.

6.2. Zoom-lens calibration

In this section we briefly describe the experiments conducted to calibrate three Hitachi HP-M1 CCD cameras with H10 × 11E Fujinon active lenses that are part of the CardEye system. More experiments and analysis can be found in Ref. [17]. For the three cameras, calibration data

Table 1
Polynomial orders and network topologies along with their number of DOF used to fit the zoom and focus varying parameters

Parameter	Network		Polynomial	
	Topology	DOFs	Order	DOFs
t_z	2–5–1	21	5	21
u_0	2–2–1	9	3	10
v_0	2–2–1	9	3	10
α_u	2–5–2–1	30	6	28
α_v	2–5–2–1	30	6	28
Total		99		97

collection was human-supervised to easily quantify the performance of our approach, therefore no outliers were present. The lens of these cameras has focal length 11–110 mm, focus range ∞–1.2 m and iris range F1.9–F22. The lens has three motors for iris, zoom and focus control. However, except for the iris motor, each of the other two motors provides a position reading presented as a dc voltage in the range 0–16384 after A/D conversion. In the following, we will refer to this range in normalized presentation from 0 to 1. For the operating range, we have chosen a focus range of $0.6 \leq m_f \leq 0.9$ which corresponds roughly to a focused distance of 1.2–2.5 m. For the zoom, we have chosen a similar range of $0.6 \leq m_z \leq 0.9$ which corresponds to focal length from approximately 11 up to 20 mm. The iris is fixed at a proper opening for all cases. We used a regular 7×7 sampling of the selected zoom and focus ranges. To collect the calibration data for each camera, a 320×243 image of a checkerboard calibration pattern consisting of two perpendicular planes was captured at each sampling position. Each time 440 corners of the black squares of the calibration pattern were measured relative to a fixed 3D coordinate system. As such, the volume of collected data per camera consists of 49×440 calibration points. Half of the points acquired from each image was used for parameter calibration, while the other half was kept for validation. The 2D points of each of the training and validation sets were uniformly distributed throughout the image plane of the cameras. It is worth mentioning here that within the selected zoom and focus ranges many of the images acquired with this type of the lens had unnoticeable lens distortion effects. The few ones, especially at smaller focal lengths, that suffered from some noticeable distortion were undistorted first using a method based on straight line correction [7].

Our neural calibration approach was then applied independently to each training set per camera. The topology for each parameter network was selected experimentally as listed in the second column of Table 1. Fig. 4(a) shows the *rms* of the calibration error of one camera on training and validation sets as training of all networks proceeds in the global optimization stage. After the initial parameter formulation step, the calibration error is about 2 pixels at the start of the global optimization stage, which managed to decrease the rms of the calibration error on both the training and the validation sets to just below 0.5 pixels. The residual calibration error of the training set in pixels at each lens setting after and before the global optimization step is illustrated in Fig. 4(b). As a result of this step, the calibration error at the different lens setting is successfully reduced to near the precision of feature extraction errors in the images. The calibration of the other cameras showed similar final results.

Fig. 5 shows some of the final camera model surfaces versus zoom and focus settings. In particular, Fig. 5(a) shows that the motion of the principal point as a function of focus tends to rotational, whereas it is nearly translational as a function of zoom. This observation is mainly due to the focus and zoom mechanisms of this lens.

Since we have not imposed on the calibration procedure the fact that the aspect ratio, α_v/α_u, should be nearly constant (from our earlier experience with these cameras, it is equal to 1) across the different lens settings, we used this to assess the results of our calibration. Our results, for the

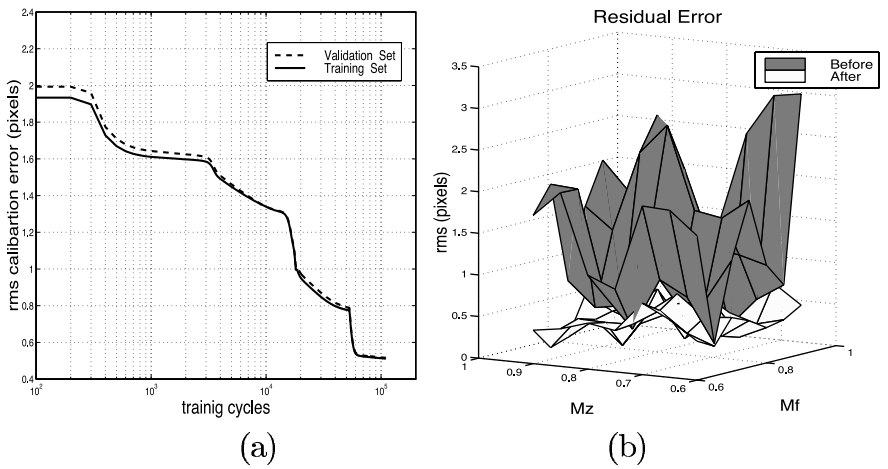

Fig. 4. The global optimization step: (a) Calibration error (in pixels) versus training cycles on training and validation sets and (b) Residual calibration error of the training set at the different lens settings after and before the global minimization step.

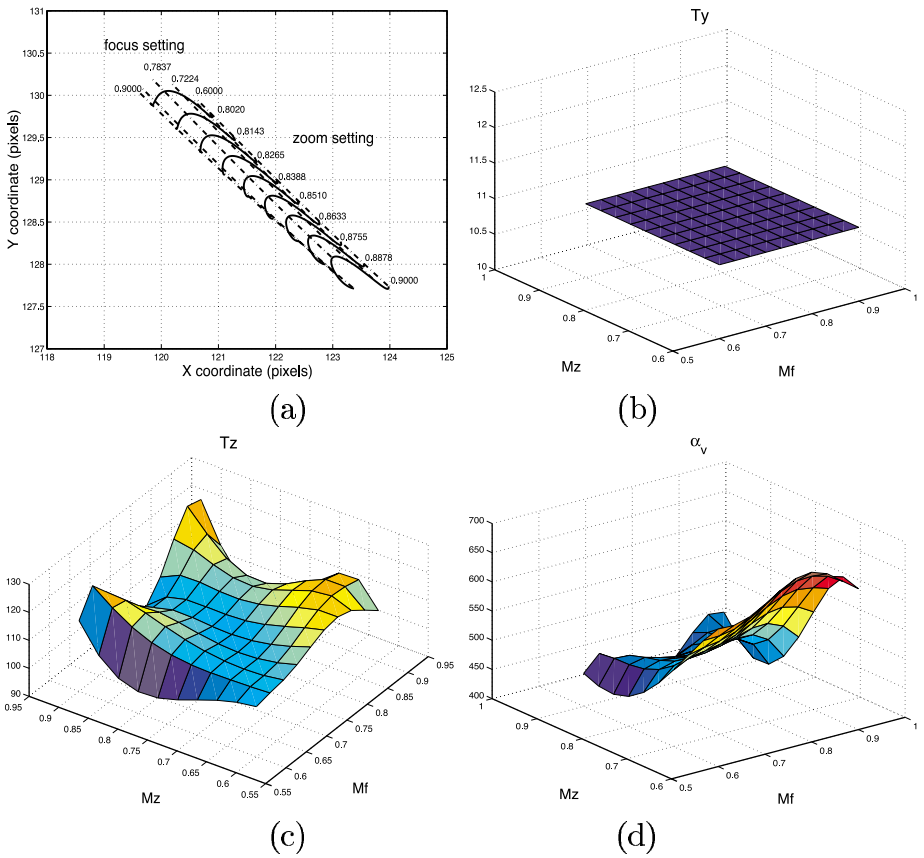

Fig. 5. Variations of some model parameters versus normalized focus and zoom settings of one lens:(a) principal point (u_0,v_0) pix., (b) t_y(cm), (c) t_z(cm) and (d) α_v(pix.).

first camera, have shown an aspect ratio with mean 1.02 and standard deviation 0.0015 across zoom and focus calibrated ranges. Similar results were obtained for the other cameras as well. This is an indication of the good calibration of these cameras.

For the sake of comparison, we applied Willson's approach [4] to the same collected calibration data of each camera. In the global optimization step using the Levenberg–Marquardt algorithm in Willson's approach, the sequence of fitting the parameter polynomials to the data affects the final calibration error. So a greedy algorithm was used to find the best sequence. The orders of the bivariate polynomials found to better describe the different model parameters are listed in Table 1. As shown in the table, the total number of degrees of freedom (DOF) used in parameter formulations is almost the same for both Willson's approach and the proposed neural approach. Note that the DOFs of a particular network topology include a fixed-bias neuron in each network layer.

The performances of both approaches on the validation sets are compared in Table 2 for the three cameras before and after the global optimization step. The tabulated rms of the calibration errors clearly shows that the neural approach did a better job in capturing the variations of the camera parameters across the zoom and focus ranges, although both approaches have used almost the same number of DOFs.

To test the generality and the repeatability of the calibrated camera model, two more experiments have been conducted. First, another set of calibration data was collected for the first camera using an identical procedure to the one used before with the pose of the camera kept unchanged from the previous one. The set had almost the

Table 2
Comparison of the rms of the calibration error in pixels before and after global optimization between our proposed approach and Willson's approach

Approach	Camera 1		Camera 2		Camera 3	
	Initial	Final	Initial	Final	Initial	Final
Willson's	2.41	1.02	1.21	0.91	1.43	0.85
Neural	1.98	0.49	1.14	0.61	1.24	0.53

Table 3
Zoom-unrelated parameters for two independent data sets

Parameter	Set 1	Set 2
t_x (cm)	8.034009	8.759913
t_y (cm)	11.410815	11.416408
R_x (rad)	0.085655	0.095442
R_y (rad)	0.057301	0.054063
R_z (rad)	−3.119118	−3.119235

Table 4
Two triangulation-based measures before and after global optimization: rms of 3D reconstruction error in centimeter and normalized stereo camera error

Approach	Initial		Final	
	RMSE (cm)	NSCE	RMSE (cm)	NSCE
Willson's	2.893	2.839	1.567	2.105
Neural	1.110	1.212	0.781	1.208

same size as the one used earlier to calibrate the camera. The camera was calibrated again using the new set. Table 3 shows the values of the zoom-unrelated parameters obtained for the two data sets. Similarly, the plots of the remaining parameters had very close resemblance to the ones previously obtained, see Ref. [17] for details. This indicates that the calibrated camera model generalizes across different sets of calibration data.

In the second experiment, several new images of the calibration pattern were captured using two cameras at various lens settings. Some lens settings that were not utilized earlier for calibration were intentionally used. The reason behind using the calibration pattern here was the fact that the 3D points on the pattern can be accurately measured relative a fixed 3D coordinate system. The stereo pair of images captured for the pattern at each lens settings were used to reconstruct the corners of the pattern squares in 3D. Two triangulation-based measures served as *another* quantitative assessment of calibration accuracy. The first is the usual rms reconstruction error (RMSE) in cm, while the other is suggested by Weng et al. [1] and called normalized stereo camera error (NSCE) (unitless). Weng claims that through this error measure, the performance of different calibration approaches can be quantitively evaluated and compared. The closer the NSCE to one, the better the calibration accuracy. Table 4 shows these two measures computed for 20 images for the neural and polynomial-based approaches using the calibrated parameters before and after the global optimization step.

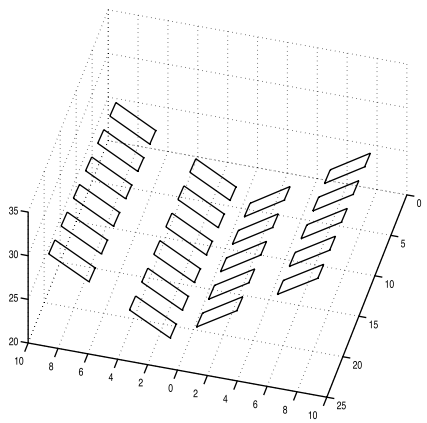

Fig. 6. Some squares of the calibration pattern reconstructed from a stereo pair acquired by two cameras.

Fig. 6 shows some of the reconstructed squares. To improve the visualization of the reconstruction, the corners of each square are connected with line segments. The coplanarity of the squares that actually lie on the same plane, their alignment and the right angles of the squares obviously reflect the quality of the calibration results.

One can conclude from Table 4 as well as Table 2 that the neural approach is more able to capture the variations of the parameters across the lens settings and it thus provides better calibration results. Moreover, the global optimization step clearly improves the calibration accuracy by tuning and coupling the different parameter formulations.

Before we conclude this section, it is worthy to make two comments on the implementation of the proposed approach. First, selecting the network topologies for the zoom-related parameters is done experimentally by examining the data. The larger the topology of the network, the better the network is able to follow variations in the data. Beyond a certain point though, these variations are due to noise rather than the underlying data. Large network topologies also have more network weights and as a result require more data (i.e. sampled lens settings) to calibrate. Given the size of the available calibration data, to select the various network topologies and to circumvent over/under-fitting, we have checked the performance of the camera calibrated model on the already formed validation set and the performance in terms of the two triangulation-based measures. The network topologies and the polynomial orders listed in Table 1 were chosen because they described better the data according to these criteria (validation set + triangulation-based measures).

The second comment is related to the run-time performance of the proposed approach. On an SGI-Indigo2 machine, the neural approach takes about 25 min while the polynomial-based approach requires near 10 min. Although the neural approach takes longer time, time can be greatly reduced if the approach is implemented in parallel since it is highly parallelizable. On the other hand, the data collection step itself often takes some hours. More importantly, this type of calibration is done off-line so accuracy not speed is the major issue.

7. Conclusions and future research

We have presented a neural framework for zoom-lens camera calibration, which produces an adjustable camera model across continuous ranges of lens control space. This framework makes use of our recently introduced neurocalibration approach [11] in order to provide a means of optimizing the model parameters over the whole calibration data. Since the data collection stage is rather tedious and time consuming, it would be advantageous to automate it. This can be done for the CardEye system through an active lighting device located at the system center. Although the calibration techniques that use calibration patterns are usually user-supervised and less likely to have outlying

data, a calibration approach using automatic data collection should be able to handle possible outliers in the data. We have demonstrated how our approach can be robust and less sensitive to outliers. We have described a number of real experiments to calibrate three zoom-lens cameras that are part of an active vision system developed in our lab. Several experiments have been conducted to evaluate and validate the calibration results. Our experimental results have demonstrated better performance of our approach compared to Willson's polynomial-based approach, which is a reference of this domain. In addition, the results have highlighted the importance of the global optimization step to account for the interaction between the different camera model parameters and to improve the calibration accuracy.

We believe that this approach has the following key features, as opposed to other techniques (e.g. [4,5,9,10]):

1. It is general; it can consider, in a straightforward manner, any number/combination of lens control parameters, e.g. zoom, focus and/or aperture.
2. Since no a priori knowledge about how lens settings affect the model parameters can be assumed available, our framework is flexible enough to capture complex variations in the model parameters across continuous ranges of control space.
3. It integrates parameter formulation with the minimization of the overall calibration error; the basis functions used for parameter formulations (which are implemented internally by the neural networks) can change from their initial forms by learning during the global optimization stage.
4. The Backpropagation-based training algorithm in the global optimization step can naturally handle an increased number of variables. As such, all of the parameters are fitted to the calibration data at the same time, while in other approaches [4,5,9,10], one parameter is fitted at a time and the final level of error generally depends on the order in which the models are fit to the data [4].

In context of neural network, this work has a number of novel aspects:

- The neurocalibration network represents a non-typical multi-layered network structure, in which the slopes of the output neurons activation functions vary as learning proceeds. Typically the activation functions of the network neurons are chosen beforehand and then kept fixed during network training.
- Each weight of the neurocalibration network has its own physical meaning since it represents a particular camera model parameter. Accordingly, each network weight may play a different role during the training of the network. Accordingly, each network weight may play a different role during the training of the network according to the available information about the camera model parameters; Some weights are allowed to update freely to optimize the network error, while others may be fixed absolutely or relatively. Usually, neural network applications treat the network as a 'black box', while network training algorithms typically view network weights as a single vector of isotropic parameters to be minimized.
- The combination of the neurocalibration network and the other MLFNs used in the global optimization stage represents a novel, non-typical neural network structure as well. We have developed an extended variant of the known Backpropagation algorithm to train these networks minimizing the overall calibration error for all the collected calibration data and yet minimizing the fitting error of each MLFN.

The future directions of this work aims at improving the accuracy and flexibility of the calibration approach. To improve the accuracy, more sampling positions during data collection are needed. We used evenly spaced samples in each of the lens control parameters. However, in order not to considerably increase the size of calibration data and without sacrificing the accuracy, one can use variable or adaptive sampling spacings, where more samples are taken in regions with high parameter variations.

Another direction worth investigating is to use one of the network pruning methods [16] to find parameter MLFNs with minimum size and yet with good generalization. In this case, each MLFN starts with a large size then it is pruned by weakening or eliminating certain weights in a selective and orderly manner while maintaining an adequate performance in terms of calibration error.

One ongoing research direction is to use this framework to model the effect of the direction of lens adjustment on the camera calibrated parameters. According to a previous, interesting study [4] using the autocollimated laser approach, the direction of adjusting the zoom or the focus of the lens has affected the trajectory of the image center with any of them, i.e. the variation of the image center with a lens control variable (focus or zoom) will be different in case of increasing the control variable from the case of decreasing it. This is mainly attributed to some amount of mechanical hysteresis in the lens system. This suggests that the lens setting-varying parameters may have different behavior depending on the direction of adjusting the lens settings. To overcome this phenomenon which would complicate the calibration process, any lens setting should consistently be approached from one direction. However, this would introduce some delays in the on-line operation of the system. Moreover, although this will increase considerably the volume of calibration data to be collected, we believe that considering this effect within the same calibration framework will improve the overall accuracy and usefulness of the adjustable zoom-lens camera model.

Acknowledgments

This research was partially supported by grants from NSF (ECS 9505674) and The Department of Defense (USNV N00014-97-11076).

References

[1] J. Weng, P. Cohen, M. Herniou, Camera calibration with distortion models and accuracy evaluation, PAMI 14 (10) (1992).
[2] M. Li, J.-M. Lavest, Some aspects of zoom lens camera calibration, PAMI 18 (1996).
[3] R. Swaminathan, S. Nayar, Non-metric calibration of wide-angle lenses and polycameras, PAMI 22 (10) (2000) 1172–1178.
[4] R.G. Willson, Modeling and calibration of automated zoom lenses, PhD Dissertation, Dept. Elect. Comp. Eng., Carnegie Mellon University, 1994.
[5] A. Wiley, K. Wong, Geometric calibration of zoom lenses for computer vision metrology, Photogrammetric Eng. Remote Sens. 61 (1) (1995).
[6] F. Hornik, Multilayer feedforward networks are universal approximators, Neural Networks 2 (1989).
[7] B. Prescott, G. McLean, Line-based correction of radial lens distortion, Graph. Models Img. Process. 59 (1) (1997).
[8] K. Tarabanis, R. Tsai, D. Goodman, Calibration of a computer controlled robotic vision sensor with a zoom lens, CVGIP 59 (2) (1994).
[9] W. Seales, D. Eggert, Active-camera calibration using iterative image feature localization, Proc. Conf. Analysis of Images and Patterns, Prague, September 1995.
[10] S. Shih, Y. Hung, W. Lin, Calibration of an active binocular head, IEEE Trans. Man, Syst. Cybernet. 28 (4) (1998).
[11] M. Ahmed, E. Hemayed, A. Farag, Neurocalibration: a neural network that can tell camera calibration parameters, Proc. ICCV, Korfu, Greece, September 1999.
[12] P. Rousseeuw, A. Roy, Robust Regression and Outlier Detection, Wiley, New York, 1987.
[13] P. Huber, Robust Statistics, Wiley, New York, 1981.
[14] F. Hampel, E. Ronchetti, P. Rousseeuw, W. Stahel, Robust Statistics: The Approach Based on Influence Functions, Wiley, New York, 1986.
[15] R. Hecht-Nielsen, Neurocomputing, Addison-Wesley, Reading, MA, 1990.
[16] S. Haykin, Neural Networks: A Comprehensive Foundation, second ed., Prentice-Hall, Englewood Cliffs, NJ, 1999.
[17] M.T. Ahmed, Zoom-lens camera calibration for an active vision system, PhD Dissertation, University of Louisville, Louisville, KY, 2001; http://www.cvip.uofl.edu/Publications.
[18] P. Meer, D. Mintz, A. Rosenfeld, D.Y. Kim, Robust regression methods for computer vision: a review, Int. J. Comput. Vis. 6 (1) (1991) 59–70.