Application of neural networks to inverse lens distortion modelling

Jason de Villiers
Council for Scientific and Industrial Research
Pretoria, South Africa
Email: jdvilliers@csir.co.za

Fred Nicolls
Department of Electrical Engineering
University of Cape Town
Cape Town, South Africa

***Abstract*—The accurate and quick modelling of inverse lens distortion to rectify images or predict the image coordinates of real world objects has long been a challenge. This work investigates the use of artificial neural networks to perform this modelling. Several architectures are investigated and multiple methods of training the networks are used to achieve the best results. The error is expressed as a physical displacement on the imaging chip so that a fair comparison can be made between other published results which are based on cameras with different resolutions. It is shown that the novel application of locally optimised neural networks to residual lens calibration data yields an inverse distortion modelling that is highly competitive with prior published results.**

I. Introduction

A. Lens Distortion

Lens distortion is that non-linear bending of light rays by a lens such that it deviates from the pin hole camera projection model. Typically complex lenses, made of several glass elements, exhibit monotonically increasing contraction of the light ray bundle towards the image centre for wide angle lenses. This is known as barrel distortion. Pin cushion distortion, found in some tele-photo lenses, is the expansion of the light ray bundle as a function of angle from the optical axis. Lenses incorporating aspherical elements can be designed to have significantly less distortion, but that residual distortion is not monotonic and is known as moustache distortion.

Distortion in images affects any measurements made from those images. Straight lines may appear curved, and measurements such as angular separation of two image points, or triangulation of a point visible to multiple cameras will be adversely affected.

The classical lens distortion model is that of Brown [1], [2] and Conrady [3]. Brown's model, given by (1), allows one to calculate with the desired accuracy, the corresponding undistorted point (i.e. that which would have been produced by a pin hole camera) for any point in the distorted image. The accuracy of the model is a function of the number of radial and tangential parameters that are determined, and whether an optimal distortion centre is found. In order to create a distortion-free image however the inverse operation is required: for each (integer located) pixel in the undistorted image, the (probably non-integer located) pixel in the distorted image is required. This allows one, after interpolation in the distorted image, to generate the distortion-free image with-out gaps or holes. This process is known variously as undistortion, inverse distortion and distortion correction and additionally allows one to determine the image coordinates of a reference determined by external means such as RADAR or geo-location.

As Brown's model contains no precise inverse, various methods of approximating it have been investigated. Candocia[4] presented an alternate scale preserving distortion function, which had an analytical inverse requiring the solving of two fifth order polynomials. Mallon and Whelan[5] used a Taylor series expansion of a Brown model containing only a few coefficients. De Villiers *et al*[6] fitted a high order Brown model to map the inverse distortion by using the pseudo-undistorted points created during distortion characterisation.

$$\begin{aligned}
x_u =& x_d + (x_d - x_c)(K_1 r^2 + K_2 r^4 + \ldots) + \\
& \big(P_1(r^2 + 2(x_d - x_c)^2) + 2P_2(x_d - x_c)(y_d - y_c)\big) \times \\
& (1 + P_3 r^2 + \ldots) \\
y_u =& y_d + (y_d - y_c)(K_1 r^2 + K_2 r^4 + \ldots) + \\
& \big(2P_1(x_d - x_c)(y_d - y_c) + P_2(r^2 + 2(y_d - y_c)^2)\big) \times \\
& (1 + P_3 r^2 + \ldots)
\end{aligned} \tag{1}$$

where:

(x_u, y_u) = undistorted image point,
(x_d, y_d) = distorted image point,
(x_c, y_c) = centre of distortion,
K_n = N^{th} radial distortion coefficient,
P_n = N^{th} tangential distortion coefficient,
$r = \sqrt{(x_d - x_c)^2 + (y_d - y_c)^2}$, and
$\ldots$ = an infinite series.

B. Artificial Neural Networks

Artificial Neural Networks (ANN) are an approximation of a biological brain. The synaptic connections to other neurons are represented by input weights. The action potential on the soma is represented by the output of a function of the difference between the sum of the weighted inputs and an internal threshold. Standard texts (e.g. [7]) can be consulted for more details.

Feed forward ANNs are a standard architecture [7] wherein the neurons are arranged in layers. The input neurons each receive one of the inputs and distribute it to the next layer. Each of the neurons in the subsequent layers accept as inputs, the outputs of all the neurons in the preceding layer, this is called a fully connected network. The original method to train a network consisting of only one neuron (sometime called a perceptron) was developed by Rosenblatt in 1958[8] and extended to networks with multiple neurons in multiple layers by Bryson and Ho in 1969[9]. This is the ubiquitous backwards propagation model, which is analogous to steepest descent numerical optimisation.

Much work has been done on applying ANNs to computer vision problems, Egmont-Peterson *et al.* [10] provide a review and classification taxonomy for their application. Camera calibration falls within their pixel-level pre-processing category. Memon and Khan [11] trained an ANN to produce the 3 dimensional position of a point given the matched image coordinates of the points as observed by a stereo camera pair. Do [12] used neural nets both to model the entire system as well as to model only the deviation from the pinhole camera model. Ahmed *et al.*[13] used an ANN to determine the intrinsic and extrinsic calibration parameters of a camera, excluding lens distortion effects which were assumed either insignificant or accounted for upstream.

C. Contribution

Previous work on modelling inverse distortion did not consider neural networks. Previous work using ANNs for camera calibration either did not consider the effect of lens distortion or implicitly modelled distortion only in the direction of distorted to undistorted domains. This work investigates the applicability of modelling inverse distortion using ANNs and the residual pseudo-undistorted points created during normal lens distortion procedures.

The rest of this paper is divided as follows: First, the equipment and methods used to capture the data are described in section II. Then the architectures of the neural networks that were evaluated is discussed in section III. Section IV describes the algorithms used to train the neural networks. Section V provides the results of the training of the neural networks. Section VI discusses and places the results in context. Finally, section VII provides some concluding remarks.

II. Data Capture Method

A 46" high definition liquid crystal display (LCD) television was used to create checker patterns to provide data for the distortion characterisation. This allowed many thousands of checkers to be generated and captured, whilst their exact positions were accurately known. A Prosilica GE1600 machine vision camera with a Schneider 4.8mm Cinegon lens was used to observe the LCD screen. The LCD and camera were arranged such that the LCD was as far as possible, but still subtended the camera's entire field of view (FOV). The Schneider lens exhibited classical monotonic barrel distortion with a horizontal FOV of approximately 82°. The GE1600 had a resolution of 1600 by 1200 and 2/3 inch CCD. The ambient lighting was carefully controlled and simple background elimination performed by subtracting intermittently refreshed reference frames with a blank LCD. This ensured that erroneous reflections in the LCD did not adversely affect the data.

The checkers were found in the camera's image with subpixel accuracy using Lucchese and Mira's[14] method of finding the saddle point of the 6 coefficient second order surface of intensity versus X and Y. This surface was fitted in a least squares sense in a window centered around the initial roughly found checker position. This refined position was further refined, using the same method and centering the window around the previous refined position to ensure that the final window was indeed centered (to the nearest pixel) around the refined checker position for maximum accuracy.

The distortion characterisation, and corresponding calculation of the pseudo-undistorted points, was performed as described by de Villiers *et al.*[6] using Brown's model with five radial coefficients, three tangential coefficients and an optimal distortion center.

III. Neural Network Architectures

All the ANNs considered in this work were of the fully connected feed forward variety. The sigmoid activation function given in (2) was used for all the neurons. Each ANN was trained to provide either the distorted horizontal or vertical image coordinate when provided with the (pseudo) undistorted horizontal and vertical image coordinates, i.e. two networks were used to do the correction.

$$Output = \frac{1}{1 + e^{th - \sum_{j=1}^{n}(w_j i_j)}} \tag{2}$$

where:

$$\begin{aligned} n &= \text{the number of inputs to the neuron,} \\ w_j &= \text{the weight associated with input } j, \\ i_j &= \text{the } j\text{th input, and} \\ th &= \text{the neuron's threshold.} \end{aligned}$$

Networks with two and three hidden layers were considered. The first evaluation consisted of ANNs ranging from 3 to 5 neurons in each of three hidden layers. The second evaluation consisted of all permutations of ANNs with two hidden layers and from 2 to 19 neurons per hidden layer. The final evaluation consisted of networks with two hidden layers with between 3 and 5 neurons each. These architectures had ANNs who's weights refined using a local optimisation algorithm.

IV. Algorithm Descriptions

For the first two sets of evaluations described in Section III, a multi-start strategy was implemented. Ten networks were generated with random weights and thresholds, and were then trained using the standard backwards propagation technique[9]. An adaptive learning rate (initially set to 0.1) was employed whereby the learning rate was increased by a multiplicative factor of 1.01 (up to a maximum of 1.0) if two of the past three epochs resulted in an improvement. Similarly the

TABLE I
SCALED BEST RESULTS FOR ANNS WITH 3 HIDDEN LAYERS

Num Neurons in first hidden layer	Num neurons in second hidden layer								
	1			3			5		
	Num Neurons in 3rd layer			Num Neurons in 3rd layer			Num Neurons in 3rd layer		
	1	3	5	1	3	5	1	3	5
2	1.100	1.038	1.076	1.063	0.919	1.053	1.066	0.522	1.045
3	1.109	1.100	1.058	1.028	0.962	0.948	1.057	1.000	0.994
4	1.109	1.077	1.026	1.052	0.706	0.604	0.705	0.579	0.300
5	1.085	1.116	1.087	0.618	0.464	0.517	0.674	0.776	0.385

TABLE II
SCALED AVERAGE RESULTS FOR ANNS WITH 3 HIDDEN LAYERS

Num Neurons in first hidden layer	Num neurons in second hidden layer								
	1			3			5		
	Num Neurons in 3rd layer			Num Neurons in 3rd layer			Num Neurons in 3rd layer		
	1	3	5	1	3	5	1	3	5
2	1.132	1.534	1.093	1.542	1.033	1.455	1.354	0.969	1.247
3	1.685	1.562	1.180	1.150	1.063	1.042	1.155	1.078	1.174
4	1.421	1.335	1.358	1.283	0.989	0.983	1.012	0.967	0.825
5	1.237	1.208	1.398	0.843	0.793	0.883	0.982	0.962	0.779

learning rate was decreased by the same multiplicative factor to a minimum of 0.05 if two of the past three epochs showed a worsening. The ANNs were trained using two thirds of the data, and evaluated on the last third. As the data was captured sequentially from top left to bottom right of the camera's FOV, the training and evaluation data were interleaved, specifically the ANNs were trained on samples $3n + 0$ and $3n + 1$ and evaluated on $3n + 2$. All the image coordinates were scaled by the imager's resolution to be in the normalized range of $(0, 1)$. This implies that the distorted image coordinates were $\in (0, 1)$ but undistorted ones corresponding to distorted points on the periphery were not.

The backwards propagation algorithm iterates sequentially through each input-output set in the epoch and makes a small adjustment. For the third series of evaluations, it was decided to determine what adjustments should be made to the neuron's parameters to make a global improvement for the entire epoch. To do this, ANNs of the specified architecture were trained and discarded using the procedure just described, until one achieved an error of 0.002 or better. Thereafter, the Leapfrog local optimiser[15] was used (with a centered difference gradient estimation) to refine the weights. This implies 2 evaluations of the epoch per weight and threshold in order to determine what step to take to improve the network. 102 epochs are thus required for a network with two inputs, two hidden layers of 5 neurons and a single output. The metric that was minimized was the Root of the Mean Square (RMS) error of the difference between the actual distorted image ordinate and the one that the ANN produced based on the pseudo-undistorted image coordinate.

V. RESULTS

Table I provides the best results achieved for each of the ten attempts for each of the ANNs with 3 hidden layers. Only ANNs to correct the horizontal lens distortion component were trained. The values are the RMS error of the ANN multiplied by a factor of 100, this scaling is common for Tables I through V and was implemented to conserve space and show only the significant digits. Table II provides the average results of each architecture's ten runs.

Fig. 1. Original undistorted image

The best results for a subset of the 2 hidden layer architecture ANNs that were trained solely via backward propagation are provided by table III. Table IV lists the average results for these architectures.

TABLE III
SCALED BEST RESULTS FOR ANNS WITH 2 HIDDEN LAYERS

Neurons in 1st layer	Neurons in 2nd Layer				
	3	7	11	15	19
3	0.730	0.770	0.946	0.972	0.936
7	0.256	0.248	0.373	0.317	0.332
11	0.212	0.247	0.344	0.351	0.315
15	0.226	0.240	0.231	0.363	0.390
19	0.223	0.226	0.281	0.361	0.421

(a) Image undistorted as per [6].

(b) Typical ANN corrected image.

(c) Image undistorted by an ANN with 11 pixel error.

(d) Image undistorted by an ANN with sub-pixel error.

Fig. 2. Distortion corrected images

TABLE IV
SCALED AVERAGE RESULTS FOR ANNS WITH 2 HIDDEN LAYERS

Neurons in 1st layer	Neurons in 2nd Layer				
	3	7	11	15	19
3	0.971	0.960	0.979	0.997	1.005
7	0.467	0.426	0.542	0.598	0.602
11	0.379	0.331	0.462	0.555	0.541
15	0.364	0.325	0.452	0.498	0.513
19	0.392	0.346	0.456	0.499	0.522

TABLE V
SCALED BEST RESULTS FOR OPTIMISED ANNS

Neurons 1st layer	Neurons in 2nd Layer					
	3		4		5	
	X	Y	X	Y	X	Y
3	0.224	0.145	0.064	0.169	0.064	0.144
4	0.059	0.055	0.122	0.065	0.068	0.063
5	0.043	0.065	0.034	0.062	0.075	0.058

The best results achieved for the two hidden layer architecture ANNs whose backwards propagation determined weights were further refined by a local optimiser are given in table V. Results for both X and Y distortion corrections are provided. Figure 2 gives a more intuitive representation of the results. Figure 1 is a raw image captured with the Prosilica GE1600 and Schneider Cinegon 4.8mm lens. Figure 2(a) is the undistorted image, calculated as described by de Villiers *et al.* [6]. A typical image undistorted by an ANN is given in figure 2(b). An image corrected by an ANN pair which converged to an accuracy of 10.98 pixels is given by figure 2(c). The image corrected by the best X and Y ANNs, which together yield an accuracy of 0.85 pixels, is provided in figure 2(d).

VI. Discussion of results

A visual comparison between figures 2(a) and 2(d), produced by the sub-pixel accurate ANN, show no perceptible differences for the inner images. Both appear to straighten all the curvature apparent in figure 1, although the ANN does perform poorly for regions corresponding close to the periphery of 1. This is due to the scaling used, where the full output range $(0, 1)$ was used to represent the image resolution, this also forces points that are not visible to the camera to be mapped onto the border of the image. Figure 2(c), created using an ANN which obtained a poorer fit shows residual curvature is still apparent even in the centre of the image, and the smearing in the periphery noticeably worse.

The values given in tables I through V are normalized results, taking into account the scaling, the X values need to be multiplied by 16, and the horizontal values by 12 to get the pixel values. However, comparing errors of distortion correction in terms of pixels is misleading as this is affected both by the resolution and the size of the imager. So a comparison in terms of microns on the imager makes more sense if one assumes similar amounts of distortion induced by the lens.

Table VI provides this comparison. The first two rows correspond to the best results achieved for ANNs with 3 and 2 hidden layers respectively, the Y distortion correction error was assumed to equal to the X error. The third row gives the results achieved with the locally optimised ANNs, and the fourth row lists the results of using high order brown models to characterise the inverse distortion as per [6]. The final rows provide, for comparison, results published in literature. Mallon and Whelan's camera resolution was inferred from their paper and the imager size assumed to be 2/3 inch.

TABLE VI
Distortion results comparison

Correction Type	Resolution		Pixel Size (μ)	Pixel Error	Micron Error
	Horiz	Vert			
ANN 3H	1600	1200	5.5	6.78	37.3
ANN 2H	1600	1200	5.5	4.80	24.75
ANN Opt	1600	1200	5.5	0.85	4.68
de Villiers[6]	1600	1200	5.5	0.30	1.65
Mallon[5]	1024	768	8.6	0.42	3.61
de Villiers[6]	667	502	13.15	0.013	0.17

It can be seen that from table VI that the optimised ANNs perform comparably well to the best models available in literature. On the same camera and lens the reverse Brown models performed only 3 times better. It achieved very similar results to Mallon and Whelan's camera which had a 6.0mm focal length and may thus have exhibited less initial distortion. Their quoted error metric is the mean, whereas the other values are RMS errors, whose magnitude is, by definition, larger.

In general the ANNs with 3 hidden layers performed better as the number of neurons increased in each layer. The performance of the ANNs with 2 hidden layers was in general superior to that of the 3 hidden layer ANNs and yielded the best results with 11 or more neurons in the first layer, and 7 or fewer in the second. The results achieved by locally optimising small ANNs with only a few neurons in 2 layers yielded results improved by almost an order of magnitude.

No scaling was performed other than the normalization required by the activation function of the neurons; results in literature ([6]) report separate scaling for each modelled parameter being required. Further work with a scaling such that valid distorted image coordinates occupy a subset of the full output range (e.g. $(0.1, 0.9)$ as opposed to $(0, 1)$) may improve the results significantly.

VII. Conclusions

It has been shown that artificial neural networks can effectively undistort an image by modelling the intractable inverse distortion. The results achieved are comparable to the most accurate algorithms when they are performed on the same apparatus. Comparing to published results, when converting errors to microns on the imaging chip, shows similar performance to established methods for modelling inverse distortion.

ANNs are thus an acceptable way to rectify all but the outermost peripheries of a distorted image, and a new scaling method that may overcome even this limitation has been proposed.

References

[1] D. Brown, "Decentering distortion of lenses," *Photogrammetric Engineering*, vol. 7, pp. 444–462, 1966.

[2] D. Brown, "Close range camera calibration," *Photogrammetric Engineering*, vol. 8, pp. 855–855, 1971.

[3] A. Conrady, "Decentered lens systems," *Monthly Notices of the Royal Astronomical Society*, vol. 79, pp. 384–390, 1919.

[4] F. Candocia, "A scale-preserving lens distortion model and its application to image registration," in *Proceedings of the 2006 Florida Conference in Recent Advances in Robotics*, ser. FCRAR 2006, vol. 1, 2006, pp. 1–6.

[5] J. Mallon and P. Whelan, "Precise radial un-distortion of images," in *Proceedings of the 17th International Conference on Pattern Recognition*, ser. ICPR 2004, vol. 1, 2004, pp. 18–21.

[6] J. de Villiers, F. Leuschner, and R. Geldenhuys, "Centi-pixel accurate real-time inverse distortion correction," in *Proceedings of the 2008 International Symposium on Optomechatronic Technologies*, ser. ISOT2008, vol. 7266, 2008, pp. 1–8.

[7] M. Negnevitsky, *Artificial Intelligence, A guide to intelligent systems*. Harlow, England: Pearson Education, 2002.

[8] F. Rosenblatt, "Perceptron simulation experiments," in *Proceedings of the institute of radio engineers*, vol. 48, 1958, pp. 301–309.

[9] A. Bryson and Y. Ho, *Applied optimal control*. New York, USA: Blaisdell, 1969.

[10] M. Egmont-Peterson, D. de Ridder, and H.Handels, "Image processing with neural networks - a review," *Pattern Recognition*, vol. 35, pp. 2279–2301, 2002.

[11] Q. Memon and S. Khan, "Camera calibration and three-dimensional world reconstruction of stereo-vision using neural networks," *International Journal of Systems Science*, vol. 32, no. 9, pp. 1155–1159, 2001.

[12] Y. Do, "Application of neural networks for stereo-camera calibration," in *Proceedings of the some or other conference*, vol. 2, 1996, pp. 13–69.

[13] M. Ahmed, E. Hemayed, and A. Farag, "Neurocalibration: A neural network that can tell camera calibration parameters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 79, pp. 384–390, 1999.

[14] L. Lucchese and S. Mira, "Using saddle points for subpixel feature detection in camera calibration targets," in *Proceedings of the Asia-Pacific Conference on Circuits and Systems*, vol. 2, 2002, pp. 191–195.

[15] J. Snyman, "An improved version of the original leap-frog dynamic method for unconstrained minimization: Lfop1(b)," *Applied Mathematics and Modelling*, vol. 7, pp. 216–218, 1983.