

# Camera distortion self-calibration using the plumb-line constraint and minimal Hough entropy.

Edward Rosten · Rohan Loveland

the date of receipt and acceptance should be inserted later

**Abstract** In this paper we present a simple and robust method for self-correction of camera distortion using single images of scenes which contain straight lines. Since the most common distortion can be modelled as radial distortion, we illustrate the method using the Harris radial distortion model, but the method is applicable to any distortion model. The method is based on transforming the edgels of the distorted image to a 1-D angular Hough space, and optimizing the distortion correction parameters which minimize the entropy of the corresponding normalized histogram. Properly corrected imagery will have fewer curved lines, and therefore less spread in Hough space. Since the method does not rely on any image structure beyond the existence of edgels sharing some common orientations and does not use edge fitting, it is applicable to a wide variety of image types. For instance, it can be applied equally well to images of texture with weak but dominant orientations, or images with strong vanishing points. Finally, the method is performed on both synthetic and real data revealing that it is particularly robust to noise.

**Keywords** Radial distortion · Camera distortion · Plumb-line constraint

---

Edward Rosten · Rohan Loveland  
Los Alamos National Laboratory  
Los Alamos, New Mexico, USA  
E-mail: edrosten@lanl.gov · loveland@lanl.gov

Rohan Loveland  
Department of Engineering Science  
University of Oxford, Oxford, UK  
E-mail: rohan@robots.ox.ac.uk

## 1 Introduction

### 1.1 Camera Calibration and Distortion Correction

Camera calibration addresses the problem of finding the parameters necessary to describe the mapping between 3-D world coordinates and 2-D image coordinates. This can be divided into the determination of extrinsic and intrinsic parameters. The extrinsic parameters are necessary to relate an arbitrary 3-D world coordinate system to the internal 3-D camera coordinate system, where the z-axis is typically taken to be along the optical axis. The intrinsic parameters are independent of the camera position, and typically include a focal length, an offset relating pixel coordinates center to the optical center, a scale factor describing the pixel aspect ratio, and one or more parameters describing non-linear radial and tangential distortions. Detailed examples of this process can be found in [13, 16], among others.

In this work we address the determination of the non-linear distortion parameters, with the assumption that the other intrinsic parameters, usually addressed assuming a pin-hole camera model, can be found after the initial non-linear distortion correction. In order to do this a model must be selected, as well as a method of estimating the model parameters.

### 1.2 Summary of Related Work

#### 1.2.1 Distortion Correction Models

A general equation for distortion correction is given by

$$I_1(\mathbf{x}') = I_0(\mathbf{x}), \quad \mathbf{x}' \equiv \mathbf{D}(\mathbf{x}), \quad (1)$$

where  $I_1$  is the output (corrected) image,  $I_0$  is the input (distorted) image, and  $\mathbf{x}$  is a pixel location in the input image

that is mapped to location  $\mathbf{x}'$  in the output image by distortion correction function  $\mathbf{D}(\mathbf{x}) : \mathbb{R}^2 \Rightarrow \mathbb{R}^2$ .

If the distortion is assumed to be isotropic, and strictly radial, then  $\mathbf{D}(\mathbf{x})$  can be written as

$$\mathbf{D}(\mathbf{x}) = f(\rho)\hat{\mathbf{r}} + \mathbf{c}, \quad (2)$$

where  $\mathbf{c}$  is the center or radial distortion, the radius vector  $\mathbf{r} = [r_1 \ r_2]^T = \mathbf{x} - \mathbf{c}$ , the radius  $\rho = \|\mathbf{r}\|$ , the normalized radius vector  $\hat{\mathbf{r}} = \mathbf{r}/\rho$  and  $f$  is a scalar function of  $\rho$ .

Several models for distortion correction have been used previously. In terms of  $f$ , the models include the polynomial model [13, 2], where

$$f(\rho) = \rho(1 + k_1\rho^2 + k_2\rho^4 + k_3\rho^6 + \dots), \quad (3)$$

and the Harris model [27], where

$$f(\rho) = \frac{\rho}{\sqrt{1 + \gamma\rho^2}}, \quad (4)$$

More complex models such as [14] include additional tangential distortion

$$\mathbf{D}(\mathbf{x}) = \begin{bmatrix} p_1(\rho^2 + 2r_1^2) + 2p_2r_1r_2 \\ 2p_1r_1r_2 + p_2(\rho^2 + 2r_2^2) \end{bmatrix} + f(\rho)\hat{\mathbf{r}} + \mathbf{c}, \quad (5)$$

to account for decentering of the optical system (where  $p_1$  and  $p_2$  are the parameters). More recently the rational function model, where polynomial and perspective transforms are combined into a rational polynomial [5, 19] has been proposed.

All of these models can have adequate performance depending on the characteristics of the camera optics. In the following we limit our discussion and implementation to a single model in order to bound the scope of the paper, though it should be noted that the method we present is extensible to any model. The Harris model is chosen for its reasonable performance, single parameter, and ease of inversion (achieved by simply negating  $\gamma$ ).

### 1.2.2 Parameter Estimation Methods

A variety of methods exist for estimating camera distortion correction model parameters. Earlier efforts relied on imagery with artificially created structure, either in the form of a test-field, populated with objects having known 3-D world coordinates, or using square calibration grids with lines at constant intervals [13, 16, 2]. Alternative approaches do not require artificially created structure, but used multiple views of the same scene. The calibration technique makes use of constraints due to known camera motion (for instance rotation) [23], known scene geometry such as planar scenes [21] or general motion and geometry constrained with the epipolar constraint [24, 1, 5].

These approaches required access to the camera in order to perform a specific operation, such as acquiring views

from multiple positions or views of a particular scene. This is problematic in instances where access is no longer available to the camera, but only to the resulting imagery.

As a result, a number of methods have been developed which can operate on single views but which make use of common structure in images such as vanishing points [3], higher-order correlations in the frequency domain [11] and straight lines [4].

The methods relying on the existence of straight lines, the ‘‘plumb-line’’ methods, were pioneered by Brown [4]. Brown used a test-field with actual strung plumb-lines, but in general these methods do not require knowledge of the locations of the lines. In particular, white plumb-lines were strung across a black background, with the plumb-bobs immersed in oil for stability. The photographic plate was exposed twice with the camera rotated about the optic axis by  $90^\circ$ , giving a nominally square grid of lines. A number of points along the lines recorded on the photographic plate were measured using a microscope (a Mann comparator). Calibration was then performed by minimizing the least-squares error in the image between distorted straight lines and the measured points on the photographed plumb-lines.

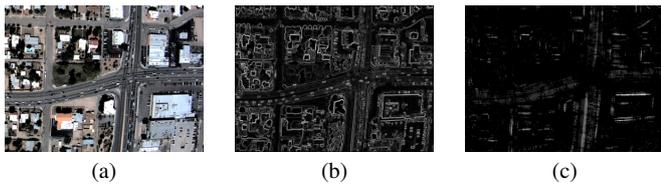
Plumb-line methods are applicable to many image types, because nearly all scenes containing man-made structures have a large number of straight lines. Plumb-line methods generally rely on the process of optimizing the distortion correction parameters to make lines that are curved by radial distortion straight in the corrected imagery. The lines can be manually selected, as in [26], or they can be found automatically, as in [7, 25, 6] by detecting edgels and linking them in to line segments.

The objective function for optimization can be formulated by undistorting the line segments and measuring the straightness by fitting a straight line [26, 7]. Alternatively, the distortion model can be chosen so that straight lines become specific family of curves, such as conics [6] or circles [25]. The distortion can then be found by fitting these curves to the distorted line segments.

## 2 Algorithm Overview

We propose a method that is simple and robust to high levels of noise, as shown in the results section. In our algorithm we calculate all image edgels, and then transform these into a one-dimensional Hough space representation of angle. This creates an orientation histogram of the edgel angles. In this form, curved lines will be represented at a variety of angles, while straight lines will be found only at one. Therefore, we optimize the model distortion parameters which minimize the entropy (or spread) of the Hough space angular representation. The individual steps are:

1. Find salient image edgels, with normal vectors.



**Fig. 1** Salient edge detection using tensor voting. (a) The original image, (b) the gradient magnitude, (c) the edge saliency.

2. Perform a distortion correcting transformation to the edgels.
3. Compute the 1-D angular Hough transform.
4. Compute an objective function defined as the spread (entropy) of the 1-D Hough transform.
5. Optimize the transformation parameters to minimize the entropy/spread based objective function, iterating steps 2–4,
6. Use the optimized transform parameters to map the input image to a corrected output image.

Note that step 1, finding the edgels, is only required once, due to the fact that the edgels, rather than the underlying image, can be transformed. This, and the other steps of the process, are described in further detail in the following respectively enumerated subsections.

### 2.1 Salient edge extraction

The structure that we are making use of in this paper consists of long, straight edges, which appear as long, smoothly curved edges in the input image.

Simply performing an edge detection does not result in these features being dominant, so to enhance the long edges, we use tensor voting [15] on the dense gradient image. The gradient is produced by finite differences, and voting is made in proportion to the gradient magnitude. We use the standard kernel for smooth, circular curves, specifically the kernel given by the equations on page 60 of [15].

For edges, each pixel is represented by a  $2 \times 2$  positive semi-definite matrix with eigenvalues  $\lambda_1$  and  $\lambda_2$ . We wish to find points which are sufficiently ‘edgy’ so we use the edge saliency function  $\phi$ , given by:

$$\phi = \max \lambda_1, \lambda_2 - e \min \lambda_1, \lambda_2. \quad (6)$$

This detects points where the edge in the main orientation is significantly brighter than other edges. We have found that  $e = 2$  produces good results. An example of these are shown in Figure 1. The normal of these edgels is given by the eigenvector corresponding to the largest eigenvalue.

The final stage is to discard the non-salient edges by thresholding at  $\phi = 0$ . For computational efficiency of the later stages, the set of edges passing the threshold is subsampled. Typically, we keep 100,000 edgels. In order to prevent the result being dominated by one region with strong edges,

we split the image up into a regular grid, and perform sub-sampling independently within each grid cell, so that each cell contains the same number of edgels.

## 2.2 Distortion correction model

### 2.2.1 Anisotropic extension

As mentioned previously, for the basic radial distortion correction model, we have based our distortion correction model on the Harris model defined in Equation 4. In order to extend the model’s flexibility, we add an anisotropic component, so that the model becomes:

$$\mathbf{D}(\mathbf{x}) = \hat{\mathbf{r}}f(\rho)(1 + g(\theta)) + \mathbf{c}, \quad (7)$$

where  $f(\cdot)$  and  $g(\cdot)$  are the distortion functions.

In particular, we define the anisotropy function,  $g(\theta)$  to be:

$$\begin{aligned} g(\theta) &= a_1 \sin(\theta + \psi_1) + a_2 \sin^2(\theta + \psi_2) + \dots \\ &= b_1 \sin \theta + b_2 \cos \theta + (b_3 \sin \theta + b_4 \cos \theta)^2 + \dots, \end{aligned} \quad (8)$$

which can be rearranged as a Fourier series. This formulation conveniently avoids the use of trigonometric functions, since  $\cos \theta = \hat{r}_1$  and  $\sin \theta = \hat{r}_2$ , where  $\hat{\mathbf{r}} = [\hat{r}_1 \ \hat{r}_2]^T$ .

### 2.2.2 Edgel transformation

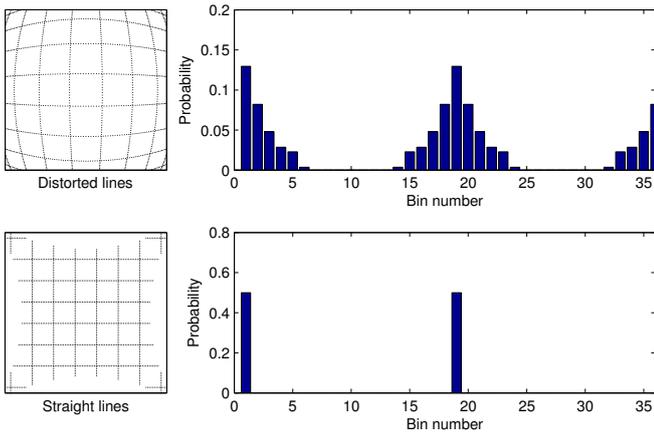
In order to evaluate the cost function it is necessary to determine the edgels’ orientation, as discussed in the previous section. If it were necessary to regenerate these by computing a completely new image for each new set of distortion model parameters, however, the optimization process (involving numerous evaluations of different sets of model parameters) would be prohibitively slow. Furthermore, errors due to resampling would be introduced when the image was undistorted at each iteration.

Fortunately, the edgels need to be determined only once, from the input image, due to the fact that the distortion correcting transformations can be applied directly to the edgels, provided that the Jacobians of the transformations are known. In order to see this, consider an edgel in the input image as part of a parameterised curve  $\mathbf{l}(t)$  in  $\mathbb{R}^2$ . The distortion correction transformation,  $\mathbf{D}$ , can be applied to this space to create a new line:

$$\mathbf{m}(t) = \mathbf{D}(\mathbf{l}(t)). \quad (9)$$

The line tangents are then found by differentiating to be:

$$\begin{bmatrix} \frac{\partial m_1}{\partial t} \\ \frac{\partial m_2}{\partial t} \end{bmatrix} = \mathbf{J} \begin{bmatrix} \frac{\partial l_1}{\partial t} \\ \frac{\partial l_2}{\partial t} \end{bmatrix}, \quad (10)$$



**Fig. 2** Images of curved and straight line images and the corresponding 1-D Hough transforms.

where the Jacobian,  $\mathbf{J}$ , is given by:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial D_1}{\partial l_1} & \frac{\partial D_1}{\partial l_2} \\ \frac{\partial D_2}{\partial l_1} & \frac{\partial D_2}{\partial l_2} \end{bmatrix}. \quad (11)$$

The edgels include normals  $\mathbf{n}$  defined at discrete points. The transformed edgel normals,  $\mathbf{h}$ , can then be found using:

$$\mathbf{h} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \mathbf{J} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \mathbf{n} \quad (12)$$

Note that we do not parameterise the line with a function. The line and its normal is known (and used) only at a discrete set of points, specifically where the edgels are detected. This means that  $\mathbf{l}(t)$  and  $\mathbf{n}(t)$  can be evaluated at every value of  $t$  we require. Since the edgel detection process also provides the normals,  $\mathbf{J}$  is only a function of the distortion model, and is therefore computed analytically from the definition of  $\mathbf{D}$ . The derivation of  $\mathbf{J}$  for the Harris model is given in Appendix A.

### 2.3 1-D Angular Hough transform

The Hough transform is a technique for finding lines in images [9]. It is sufficiently well-known so that only a truncated explanation will be given here. The basis of the technique is the transform of a line to a point in ‘‘Hough’’ space.

By way of example, if a line is defined by  $y = mx + b$ , then it can be represented by a single point in a 2-D Hough space of  $[m] \times [b]$ . An edgel, which contains both a normal vector (and thus the slope of a line) and a discrete point (which a corresponding line would pass through), can also be mapped to a single point in Hough space. In practice the  $mx + b$  formulation is unwieldy, so

$$\rho = x \cos \theta + y \sin \theta \quad (13)$$

is used, where  $\rho$  is the shortest distance from the line to the origin, and  $\theta$  is defined by the vector normal to the line.

The Hough space can be quantized into discrete bins, and each edgel can be assigned to a particular bin. Given a set of edgels, support for the existence of a particular line is then indicated by the accumulation of a large number of edgels in the corresponding bin. The edgels from a curve end up in adjacent bins, resulting in a diffuse cluster of non-zero bins, while edgels that are all on the same line end up in exactly the same bin. This important fact motivates the definition of our objective function, as explained in the next section.

In general we anticipate the presence of a number of parallel lines in the scene, and would like to utilize their reinforcement of each other. We therefore marginalize the  $[\rho] \times [\theta]$  Hough space into a 1-D  $\theta$  space by summing over the  $\rho$  values for each  $\theta$ . An example of some images with curved lines and straight lines, and their corresponding 1-D Hough transforms, are shown in Fig. 2. Note that the two principal line directions are mapped to two corresponding peaks in Hough space.

### 2.4 Entropy-based objective function

The radial distortion correction method presented here is motivated by the observation that curved lines map to spread out peaks in Hough space, while straight lines map to a single bin. Therefore, it is desirable to have an objective function that measures this spread. In information theory this quality is represented by entropy [22]. We have therefore normalized the 1-D Hough representation, and treat it as a probability distribution. The objective function is then:

$$C(\mathbf{H}) \equiv - \sum_{b=1}^B p(H_b) \log_2(p(H_b)), \quad (14)$$

where  $\mathbf{H}$  is the discretized 1-D angular Hough transform of the edgels of an input image for a given set of model parameters,  $p(H_b)$  is the value of a given normalized Hough bin, and  $B$  is the number of bins. Minimizing  $C$  therefore minimizes the spread.

### 2.5 Optimization

The cost function has many local minima, so to optimize it effectively a reasonable strategy is needed. As a broad overview, we have found that the MCDH (Monte-Carlo downhill) strategy works best. Essentially, starting parameters are selected at random, and the downhill simplex algorithm [17] (specifically the variation given in [18]) is used to optimize the cost from the selected starting points. The best result is then selected. The downhill simplex algorithm is iterated

until the residual drops below a threshold ( $10^{-15}$ ), or 1000 iterations occur, whichever is sooner. The residual is computed as:

$$r = \frac{\|\mathbf{s}_{\text{best}} - \mathbf{s}_{\text{worst}}\|_2}{\|\mathbf{s}_{\text{best}}\|_2},$$

where  $\mathbf{s}_{\text{best}}$  is the location of the best (smallest) point on the simplex, and  $\mathbf{s}_{\text{worst}}$  is the location of the worst (largest point) on the simplex.

In order to implement this method effectively, several techniques are needed. Parameters computed in the Hough transform will not lie exactly at the bin centres. If the parameter is simply placed in the closest bin, then the quantization caused by this causes flat areas to exist in the cost function. The flat areas can easily confound the optimizer, so instead, the neighbouring bins are incremented using linear interpolation.

The optimizer is most effective when the parameters being optimized have similar orders of magnitude, since otherwise roundoff error will cause errors in the computation of new simplexes. In this case, a simple scale based on *a priori* knowledge of approximate parameter values suffices.

For radial distortion,  $\mathbf{c}$  is roughly in the middle of an image and is of the order  $10^2$  to  $10^4$  pixels. Given Equation 4, it is reasonable to expect  $\gamma$  to be approximately in the range  $\pm\rho_{\text{max}}^{-2}$ , where  $\rho_{\text{max}} \approx |\tilde{\mathbf{c}}|$  and  $\tilde{\mathbf{c}}$  is the centre of the image. Therefore, in the optimization, we solve for  $\beta$ , where  $\beta = 100\gamma\rho_{\text{max}}^2$ , which brings  $\beta$  in to the same range as  $\mathbf{c}$ .

Similarly, we have found that the values of the anisotropic coefficients,  $b_1, \dots$  are of order  $10^{-3}$ , so we instead solve for  $d_i$ , where  $d_i = 10^5 b_i$ . The resulting optimization is performed over the vector:

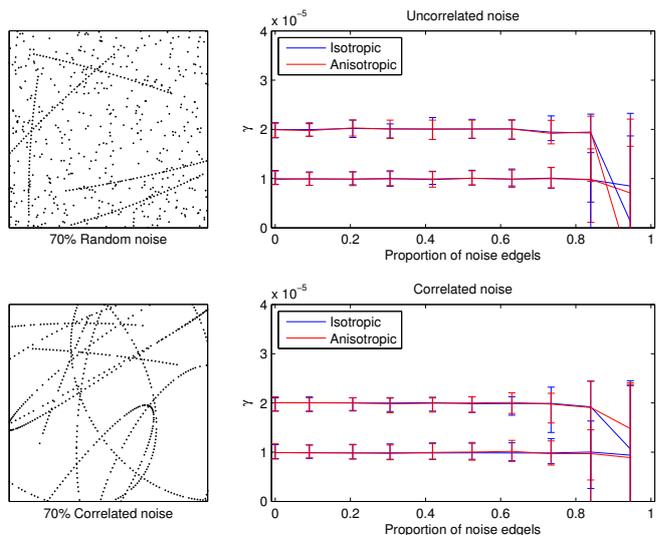
$$[c_1, c_2, \beta, d_1, d_2, d_3, d_4, d_5, d_6]. \quad (15)$$

The random selection of the starting parameters must be based around some knowledge of the values. The centre of radial distortion is usually near to the centre of the image, so we draw the initial centres from a normal distribution centred at  $\tilde{\mathbf{c}}$  with a standard deviation of  $\frac{|\tilde{\mathbf{c}}|}{20}$ . Given the approximate values of  $\gamma$ , and therefore  $\beta$ , we draw initial values of  $\beta$  from  $\mathcal{N}(0, 100)$ . The initial anisotropy parameters are set to zero. For our application and these parameters, we have found that 120 MCDH iterations is sufficient.

## 2.6 Generating the output image

In order to generate the output (corrected) image the following procedure is needed:

1. For each point,  $\mathbf{x}'$ , in a grid defined on the *output* image  $I_1$ , find the corresponding point in the distorted input image  $I_0$  given by  $\mathbf{D}^{-1}(\mathbf{x}')$ .



**Fig. 3** (Left) Illustration of the simulated data (shown with 100 points per line). (Right) Performance of the algorithm on the simulated data, with  $\gamma = 10^{-5}$  and  $\gamma = 2 \times 10^{-5}$ . The median value of gamma is shown with error bars at the 10<sup>th</sup> and 90<sup>th</sup> percentile.

2. Copy the pixel back, performing the assignment:

$$I_1(\mathbf{x}') := I_0(\mathbf{D}^{-1}(\mathbf{x}')).$$

Since images are discrete, interpolation will be required to find  $I_1(\mathbf{D}^{-1}(\mathbf{x}'))$ . Bicubic interpolation has been used to generate the results shown in the following section.

## 3 Results

### 3.1 Synthetic tests

In order to compute the sensitivity to various kinds of noise, we create a number of synthetic images, using known radial distortion levels, and compute the resulting radial distortion correction parameters using our technique. We investigate the two correction models, namely the strictly radial correction of the unmodified Harris model, and the anisotropic extension discussed previously. The images are designed to test the performance of the algorithm in the presence of measurement error in the plumb-lines, and both uncorrelated and correlated non-plumb-line data.

The  $250 \times 250$  pixel test images consist of five good quality lines (no point can be within 60 pixels of the centre of the image), consisting of 10 points on each line, drawn with barrel distortion and with random orientation errors. Then (in the case of uncorrelated noise), a number of randomly placed, randomly oriented points are added. In the case of correlated noise, a number of randomly placed, sized and oriented ellipses (with the same approximate point spacing as the lines) are added. Our algorithm is then applied to the test image to estimate the parameters.

For every selected noise proportion for the simulation, 200 test images are created for each of two different values of  $\gamma$ . The results of these, and an illustration of the simulated data is shown in Figure 3. As can be seen, the technique is very robust to noise, producing good results with up to 70% contamination and interestingly it does not perform much worse in the case of correlated noise. Furthermore, although the anisotropic extension (in this case) introduces six extra parameters to the optimization, this has no significant effect on the stability of the technique in high noise situations.

### 3.2 Example images

The results on two real images are shown in Figure 4. In the corrected images, real-world straight lines have been annotated with lines to show that there is no significant amount of curvature. The upper image shows the technique operating on an image where the lines belong to vanishing points (note that some distortion was artificially added to this image in addition to the camera distortion present). Note the presence of strong curved edges along the boundaries of the cars and parts of the building.

The lower image shows the results on an aerial image of a city. Although there are two principal directions in this image, the contrast on the principal edges is low compared to many of the other features. Additionally, the anisotropy extension was required to correct the very strong distortion in the lower left corner.

### 3.3 Comparison to other techniques

To gauge the accuracy of our method, we have compared it to a technique based on structured scenes. In particular we have used the technique of [10], which uses multiple images of a planar grid of squares to determine the intrinsic and extrinsic camera calibration parameters. This system is able to calibrate cubic, quintic, Harris and Harris with unit aspect ratio camera models. We use the last listed model, since this matches the camera model in the paper. In particular, we use this program to optimize the model:

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} f & 0 \\ 0 & f \end{bmatrix} \begin{bmatrix} x_c \\ y_c \end{bmatrix} \frac{1}{\sqrt{1 + \alpha(x_c^2 + y_c^2)}}, \quad (16)$$

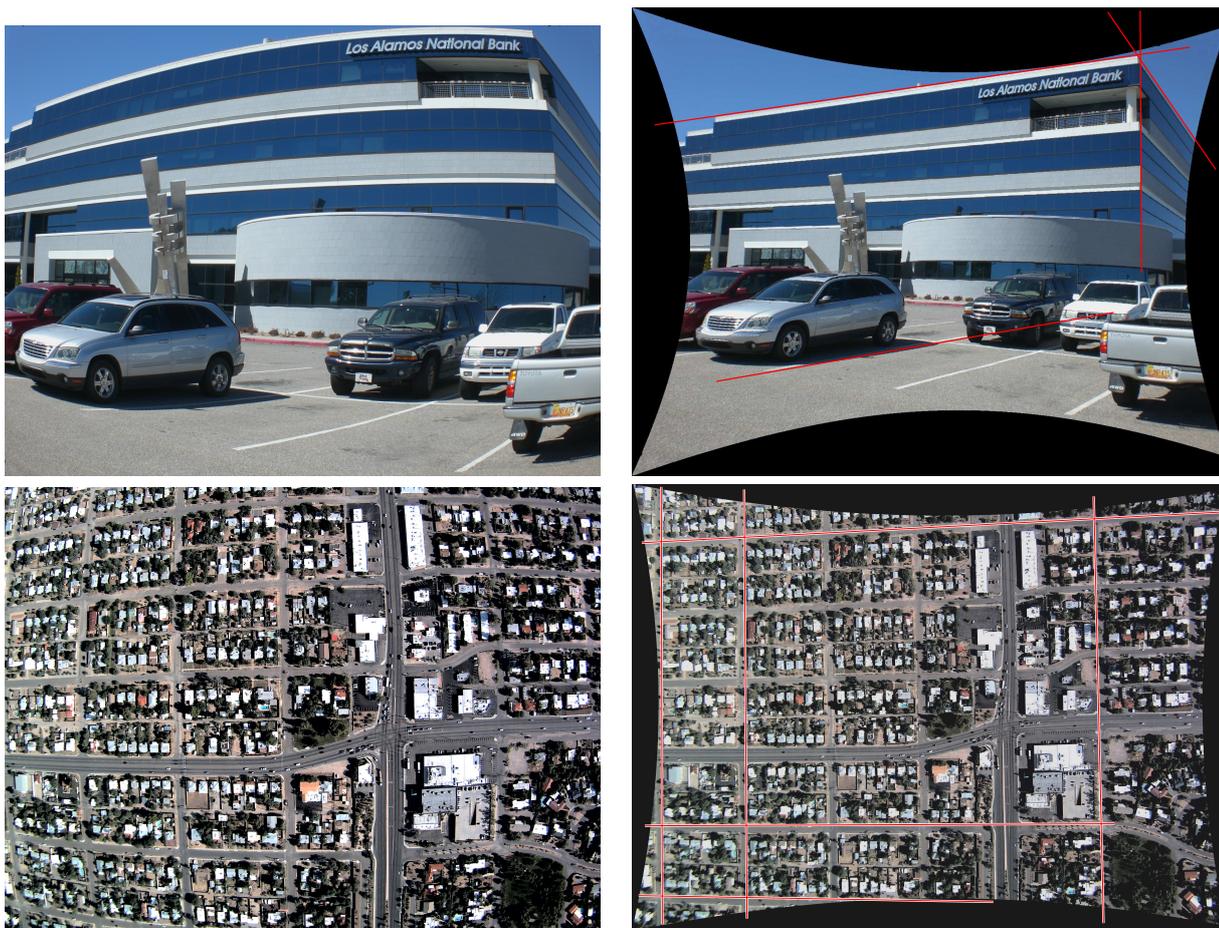
where  $(u, v)$  is the optic axis,  $(x_i, y_i)$  is a coordinate in the image, and  $(x_c, y_c)$  is a coordinate in ideal, normalized camera's image plane. Since the technique is a structured scene technique, it also estimates the focal length of the camera,  $f$ . To translate from one model to the other, set  $\gamma = \frac{\alpha}{f^2}$ . Note that this assumes that the centre of radial distortion is at the optic axis of the camera, which is a reasonable assumption.

In order to measure the quality of the camera calibration, we localise a known 3D model in an image, and measure the root mean square (RMS) error between the control points on the model and the measured points in the image. The model consists of 13 by 9 black and white squares in a chequerboard pattern on a flat plane. The model is warped using a homography (the parameters of which must be determined), then by the calibrated radial distortion function and rendered in to the image. We then search normal to the rendered edges, looking for edges in the image. We define the error to be the distance offset in pixels between the rendered model edge and the measured edge position in the image. The parameters of the homography are then adjusted using the downhill-simplex algorithm to minimize the RMS error. In particular, we search 15 pixels in either direction, and take the point with the highest gradient, measured using the kernel  $[-1 \ -2 \ -1 \ 0 \ 1 \ 2 \ 1]$ . If the highest gradient does not exceed some threshold, or is not a local maxima, then the edge search is discarded and does not contribute to the RMS error. Quadratic interpolation is then used to find the edge position to subpixel accuracy. The search distance is sufficiently small that data association is trivial, and robust fitting is not required. A more detailed treatment of very closely related systems for aligning models is given in [12, 8].

The results of this test are shown in Figure 5. As expected, the technique which uses many (in this case approximately 100) images of a structured scene performs best. The performance of our method varies somewhat with the contents of the scene, but performs well on certain images. One interesting property of our technique is well illustrated. The property is that edgels from multiple, unrelated images can be aggregated to provide more straight edges. In the case shown, the resulting calibration is superior to the calibration generated from a single image of a carefully constructed grid.

## 4 Discussion and conclusions

In this paper, we have presented a new, simple and robust method for determining the radial distortion of an image using the plumb-line constraint. The technique works by first extracting salient edgels and then minimizing the spread of a 1D angular Hough transform of these edgels. The technique is simple and because no edge fitting is performed, the technique is very robust to the presence of noise. Furthermore, the technique is more generally applicable than other plumb-line techniques in that the lines used do not need to be continuous. The technique works on textures with principal directions, as illustrated by the aerial image of a city, where the salient edge detection results in a large number of relatively small edge fragments.



**Fig. 4** (Left) Original, distorted images. (Right) Images undistorted using our technique. Some lines which are straight in the world have been annotated with straight lines in the undistorted image. The images are of a building, showing strong vanishing points and an aerial image of a city.

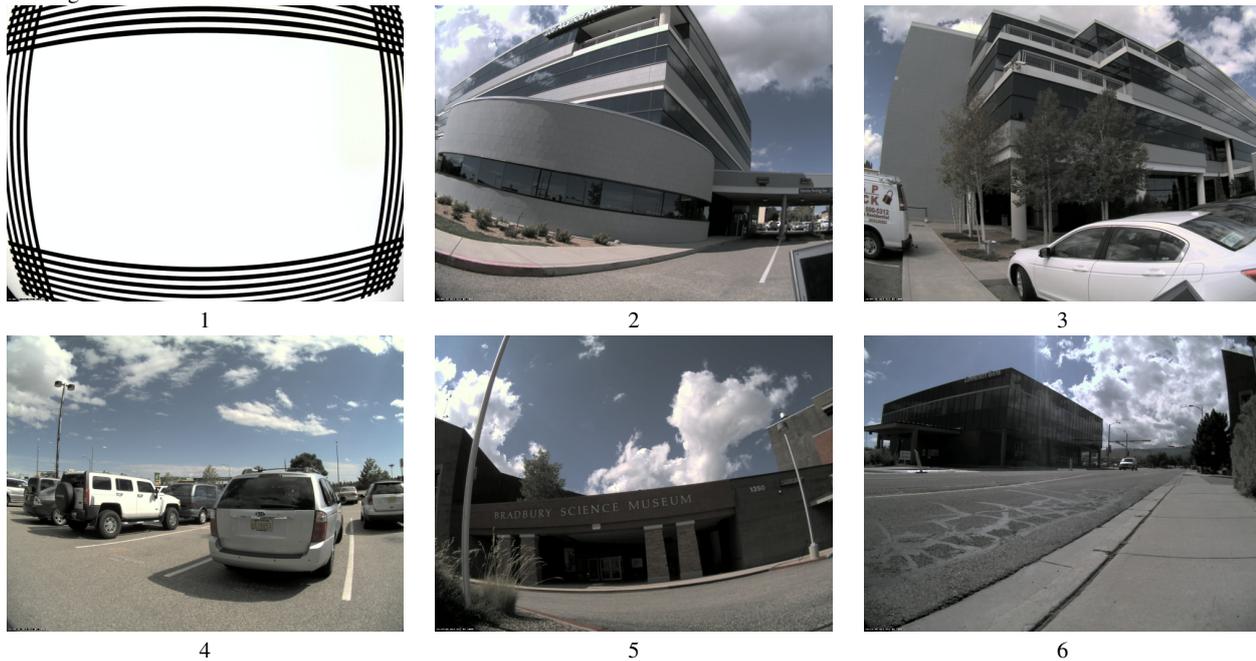
The proposed algorithm has a number of parameters: the parameters of the tensor voting kernel, the number of bins and the parameters of the optimization. In practice, the selection of these parameters are not critical, and indeed the same set of parameters was used for the simulated data, the example images and the test images shown.

Our method is flexible in that it does not impose constraints beyond the presence of one or more straight edges: it is not a requirement that the edges share vanishing points, or structure of any particular kind. It is not even a requirement that the edges belong to a related set of images. The technique can be equally applied to edges from multiple images of unrelated scenes taken with the same camera parameters. Finally, our method is widely applicable because it is, in terms of RMS error, able to produce a calibration to within three percentage points of a technique requiring access to the camera and structured scenes.

## References

1. Barreto, J.P., Daniilidis, K.: Fundamental matrix for cameras with radial distortion. In: 10<sup>th</sup> IEEE International Conference on Computer Vision, vol. 1, pp. 625–632. Springer, Beijing, China (2005)
2. Beyer, H.: Accurate calibration of ccd-cameras. IEEE Computer Society Conference on Computer Vision and Pattern Recognition pp. 96–101 (1992)
3. Bräuer-Burchardt, C., Voss, K.: Automatic correction of weak radial lens distortion in single views of urban scenes using vanishing points. In: 9<sup>th</sup> International Conference on Image Processing, vol. 3, pp. 865–868. IEEE Computer Society (2002)
4. Brown, D.C.: Close-range camera calibration. Photogrammetric Engineering **37**, 855–866 (1971)
5. Claus, D., Fitzgibbon, A.: A rational function lens distortion model for general cameras. IEEE Computer Society Conference on Computer Vision and Pattern Recognition **1**, 213–219 (2005)
6. Claus, D., Fitzgibbon, A.W.: A plumline constraint for the rational function lens distortion model. In: Proceedings of the British Machine Vision Conference, pp. 99–108 (2005)
7. Devernay, F., Faugeras, O.: Straight lines have to be straight. Machine Vision and Applications **13**(1) (2001)
8. Drummond, T., Cipolla, R.: Real-time visual tracking of complex structures. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(7), 932–946 (2002)

Test images:



Calibration results:

Calibration	$c_1$	$c_2$	$\gamma \times 10^7$	RMS error
Uncalibrated	1024	768.0	0.000	$> 40\ddagger$
libCVD camera calibration	1030	687.0	3.780	1.67
Test image 1	1075	708.6	3.700	2.38
Test image 2	1325	769.0	3.408	12.2 $\dagger$
Test image 3	1149	627.3	4.317	4.75
Test image 4	1012	739.1	2.814	6.08
Test image 5	1084	611.4	3.544	3.60
Test image 6	1226	968.0	3.405	12.8 $\dagger$
Test images 6 and 2	1047	725.6	4.075	2.21

**Fig. 5** Calibration results for an alternative method and some test images, including a calibration grid and some images of urban scenes. All images are  $2048 \times 1536$  pixels.  $\ddagger$  In the uncalibrated case, the errors are sufficiently that the search radius had to be increased to 100 pixels. This results in the RMS error being an underestimate, since for large errors, the system sometimes finds a closer edge (so the measured error is too small), or fails to find the edge at all (so large errors are not included).  $\dagger$  In some cases, the errors caused a large number of the correspondences to be missed with a search radius of 15 pixels, so it was increased to 36 pixels. For the last row of the table, the edgels from the two images were aggregated and treated as a single image. In our model,  $\mathbf{c} = [c_1 \ c_2]$  and for the libCVD model,  $u = c_1$  and  $v = c_2$ .

9. Duda, R.O., Hart, P.E.: Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM* **15**(1), 11–15 (1972)
10. Ethan Eade, *et. al.*: Camera calibration in [20]: progs/calib.cxx (2008, CVS tag SNAPSHOT\_20080904)
11. Faird, H., Popescu, A.C.: Blind removal of lens distortion. *Journal of the Optical Society of America A* **18**(9), 2072–2078 (2001)
12. Harris, C., Stennett, C.: RAPID, a video rate object tracker. In: 1<sup>st</sup> British Machine Vision Conference, pp. 73–77. British Machine Vision Association, Oxford (1990)
13. Heikkilä, J.: Geometric camera calibration using circular control points. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(10), 1066–1077 (2000)
14. Heikkilä, J., Silvén, O.: A four-step camera calibration procedure with implicit image correction. In: 11<sup>th</sup> IEEE Conference on Computer Vision and Pattern Recognition, pp. 1106–1112. Springer, San Juan, Puerto Rico (1997)
15. Lee, M., Medioni, G.: Grouping  $\cdot$ ,  $-$ ,  $\rightarrow$ ,  $\ominus$ , into regions, curves and junctions. *International Journal of Computer Vision and Image Understanding* **76**(1), 54–69 (1999)
16. Lenz, R., Tsai, R.: Techniques for calibration of the scale factor and image center for high accuracy 3-d machine vision metrology. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **10**(5), 713–720
17. Nelder, J., Mead, R.: *Computer Journal* **7**, 308–313 (1965)
18. Press, W.H., Teukolsky, S.A., Vetterling, W.H., Flannery, B.P.: *Numerical Recipes in C*. Cambridge University Press (1999)
19. Rosten, E., Cox, S.: Accurate extraction of reciprocal space information from transmission electron microscopy images. In: *Advances in Visual Computing*. LNCS 4292, vol. 1, pp. 373–382 (2006)
20. Rosten, E., Drummond, T., Eade, E., Reitmayr, G., Smith, P., Kemp, C., Klein Georg, *et. al.*: libCVD: Cambridge vision dynamics library (2008). <http://savannah.nongnu.org/projects/libcvd>
21. Sawhney, H.S., Kumar, R.: True multi-image alignment and its application to mosaicing and lens distortion correction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **21**(3), 235–243 (1999)
22. Shannon, C.M.: A mathematical theory of communication. *Bell System Technical Journal* **27**, 379–423 and 623–656 (1948)

23. Stein, G.: Accurate internal camera calibration using rotation, with analysis of sources of error. In: 5<sup>th</sup> IEEE International Conference on Computer Vision, p. 230. Springer, Boston MA, USA (1995)
24. Stein, G.: Lens distortion calibration using point correspondences. In: 10<sup>th</sup> IEEE Conference on Computer Vision and Pattern Recognition, p. 602. Springer, San Fransisco, California, USA (1996)
25. Strand, R., Hayman, E.: Correcting radial distortion by circle fitting. In: 16<sup>th</sup> British Machine Vision Conference. British Machine Vision Association, Oxford (2005)
26. Swaminathan, R., Nayar, S.: Nonmetric calibration of wide-angle lenses and polycameras. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(10), 1172–1178 (2000)
27. Tordoff, B., Murray, D.W.: The impact of radial distortion on the self-calibration of rotating cameras. Computer Vision and Image Understanding **96**(1), 17–34 (2004)

## A Derivation of $\mathbf{J}$ for the Harris model

Taking  $\hat{\mathbf{r}}$  to be a column vector (i.e.  $\hat{\mathbf{r}} = \begin{bmatrix} \hat{r}_1 \\ \hat{r}_2 \end{bmatrix}$ ):

$$\mathbf{J} = \frac{\partial \mathbf{D}}{\partial \mathbf{x}} = \frac{\partial \hat{\mathbf{r}}}{\partial \mathbf{x}} f(\rho)(1 + g(\theta)) + \hat{\mathbf{r}}(1 + g(\theta)) \frac{\partial f(\rho)}{\partial \rho} \frac{\partial \rho}{\partial \mathbf{x}} + \hat{\mathbf{r}} f(\rho) \frac{\partial g(\theta(\mathbf{x}))}{\partial \mathbf{x}} \quad (17)$$

Where:

$$\frac{\partial \hat{\mathbf{r}}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial \hat{r}_1}{\partial x_1} & \frac{\partial \hat{r}_1}{\partial x_2} \\ \frac{\partial \hat{r}_2}{\partial x_1} & \frac{\partial \hat{r}_2}{\partial x_2} \end{bmatrix} = \frac{1}{\rho^3} \begin{bmatrix} \hat{r}_2^2 & -\hat{r}_2 \hat{r}_1 \\ -\hat{r}_2 \hat{r}_1 & \hat{r}_1^2 \end{bmatrix}, \quad (18)$$

$$\frac{\partial f(\rho)}{\partial \rho} = (1 + \gamma \rho^2)^{-\frac{3}{2}}, \quad (19)$$

$$\frac{\partial \rho}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial \rho}{\partial x_1} & \frac{\partial \rho}{\partial x_2} \end{bmatrix} = [\hat{r}_1 \ \hat{r}_2], \quad (20)$$

and:

$$\frac{\partial g}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial g}{\partial x_1} & \frac{\partial g}{\partial x_2} \end{bmatrix} \quad (21)$$

The function  $g(\theta)$  is a function of  $\sin \theta$  and  $\cos \theta$ , where:

$$\cos \theta = r_1 / \rho = \hat{r}_1, \text{ and} \quad (22)$$

$$\sin \theta = r_2 / \rho = \hat{r}_2, \quad (23)$$

we can rewrite  $g(\theta(x))$  as:

$$g(\hat{\mathbf{r}}) = b_1 \hat{r}_2 + b_2 \hat{r}_1 + (b_3 \hat{r}_2 + b_4 \hat{r}_1)^2 + (b_5 \hat{r}_2 + b_6 \hat{r}_1)^3, \quad (24)$$

so,

$$\frac{\partial g}{\partial \mathbf{x}} = \frac{\partial g(\hat{\mathbf{r}})}{\partial \hat{\mathbf{r}}} \frac{\partial \hat{\mathbf{r}}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial g(\hat{\mathbf{r}})}{\partial \hat{r}_1} & \frac{\partial g(\hat{\mathbf{r}})}{\partial \hat{r}_2} \end{bmatrix} \frac{\partial \hat{\mathbf{r}}}{\partial \mathbf{x}} \quad (25)$$

where:

$$\frac{\partial g(\hat{\mathbf{r}})}{\partial \hat{r}_1} = b_1 + 2b_3(b_3 \hat{r}_1 + b_4 \hat{r}_2) + 3b_5(b_5 \hat{r}_1 + b_6 \hat{r}_2)^2 \quad (26)$$

$$\frac{\partial g(\hat{\mathbf{r}})}{\partial \hat{r}_2} = b_2 + 2b_4(b_3 \hat{r}_1 + b_4 \hat{r}_2) + 3b_6(b_5 \hat{r}_1 + b_6 \hat{r}_2)^2. \quad (27)$$

Substituting Equations 18, 19, 20, 25, 26 and 27 in to Equation 17 gives the analytic solution for the  $\mathbf{J}$ .