Automatic camera calibration from a single Manhattan image

Jonathan Deutscher, Michael Isard, and John MacCormick
Systems Research Center
Compaq Computer Corporation
130 Lytton Avenue, Palo Alto, CA 94301

Abstract. We present a completely automatic method for obtaining the approximate calibration of a camera (alignment to a world frame and focal length) from a single image of an unknown scene, provided only that the scene satisfies a *Manhattan world* assumption. This assumption states that the imaged scene contains three orthogonal, dominant directions, and is often satisfied by outdoor or indoor views of man-made structures and environments.
The proposed method combines the calibration likelihood introduced in [5] with a stochastic search algorithm to obtain a MAP estimate of the camera's focal length and alignment. Results on real images of indoor scenes are presented. The calibrations obtained are less accurate than those from standard methods employing a calibration pattern or multiple images. However, the outputs are certainly good enough for common vision tasks such as tracking. Moreover, the results are obtained without any user intervention, from a single image, and without use of a calibration pattern.

1 Introduction

There is a rich literature of methods for finding camera calibration. If the user is prepared to construct a calibration pattern and generate images which prominently display the pattern, then well-established techniques can be used to find the internal parameters of the camera (e.g. [8, 16, 18]). Alternatively, it is common in structure from motion applications to infer a camera calibration in parallel with scene geometry from a sequence of images of rigid structure — this topic is covered extensively in [9]. Some applications, in particular visual surveillance, assume a single fixed monocular camera viewing a scene, often either indoors or in a city landscape. Because the camera is fixed, structure from motion algorithms cannot be used to estimate camera calibration. On the other hand it might be impractical to generate images of a calibration grid in each surveillance camera. This paper presents a method for estimating a camera's internal calibration parameters along with its world-orientation given a single image of a somewhat structured scene. Given the limitations of basing the calibration estimate on a single image of an unknown scene, it is not expected that this method will have accuracy comparable to a method exploiting multiple images

or known grids. There are applications for which even an approximate calibration is useful, however, and these include visual tracking and image retrieval or object recognition tasks including normalising images to be upright or finding objects in the ground plane.

The Manhattan World assumption was introduced by Coughlan and Yuille [5] to model scenes in which edges lie predominantly in three orthogonal directions. They observed that the assumption frequently holds within buildings and in city streets, and used Bayesian inference to estimate the compass direction of a calibrated camera held vertically. We make use of the Manhattan framework developed in [5] and combine it with a stochastic search algorithm which performs mixture density estimation in an annealing framework [13] to estimate the focal length and alignment of an unknown camera given a single image of a Manhattan scene. This comprises a significant extension to the method of Coughlan and Yuille, since they exploited the Manhattan assumption to find a single rotation parameter only and were thus able to employ an exhaustive search algorithm which is impractical in the higher-dimensional problem we address. Because the algorithm searches for camera parameters which align with orthogonal axes in the world, it implicitly estimates the three vanishing points of those axes. The use of vanishing points for camera calibration has been studied extensively (e.g. [3, 11]). McLean and Kotturi [14] introduced a fully automatic vanishing point detection method, employing clustering techniques to obtain vanishing points from lines which are segmented from the image in a preprocessing stage. More recent approaches to calibration using vanishing points include [4, 6, 7, 12]; however these either require multiple images or manual labelling of detected lines. Another interesting approach is [1], in which vanishing points in an urban scene are estimated accurately using the EM algorithm, initialised with the output of a Hough transform. This method employs an appealing probabilistic formulation, but still requires explicit edge detection.

The advantages of the approach presented here are twofold. First, by using a likelihood model for the image together with Bayesian inference we can incorporate all of the image data without a separate edge-detection and clustering stage. Second, we exploit the orthogonality constraint explicitly to compute a joint estimate of the three vanishing points rather than trying to enforce orthogonality and estimate the internal calibration after the vanishing points have been found.

2 Bayesian calibration

We represent a camera's calibration by a parameter vector X defined in section 2.1. In principle these parameters can encode the camera's full projection matrix along with any additional information of interest, for example lens distortion parameters. In practice certain parameters such as skew and the position of the principal point can be held fixed, either because they are assumed known or because of difficulties in accurately estimating them from the available data.

The distribution $p(X|Z)$ describes the probability of a calibration X given an image Z. The problem addressed in this paper is that of finding a mode of

this distribution, i.e. a most probable candidate calibration $\hat{X}$ given Z. In order to estimate $\hat{X}$ we write

$$p(X|Z) \propto p(Z|X)p(X).$$

The Manhattan World framework [5] gives a formula for the likelihood $p(Z|X)$ which is described in section 2.2. We supply a prior $p(X)$ over calibrations and an estimation algorithm which is described in section 2.3.

2.1 Calibration parameterisation

We represent a camera's calibration by the following parameter vector

$$X = \{q_1, q_2, q_3, q_4, \phi\}$$

where the q_i comprise a quaternion defining a rotation matrix and ϕ is the focal length of the camera in pixels. We initially tried to estimate the principal point (p_x, p_y) of the camera but found that the variance in the estimates was much greater than the size of the image. Since it is well known that estimation of the principal point of a camera is very sensitive to noise [9], in the experiments below we assume the principal point is at the centre of the image. We also tried to estimate radial distortion parameters, but again found the estimates to be dominated by noise, so we assume no radial distortion.

Using standard definitions [17] the quaternion $q = (q_1, q_2, q_3, q_4)$ maps to a rotation matrix R given by

$$R = \frac{1}{|q|}\begin{pmatrix} q_1^2+q_2^2-q_3^2-q_4^2 & 2(q_2q_3-q_1q_4) & 2(q_2q_4+q_1q_3) \\ 2(q_2q_3+q_1q_4) & q_1^2-q_2^2+q_3^2-q_4^2 & 2(q_3q_4-q_1q_2) \\ 2(q_2q_4-q_1q_3) & 2(q_3q_4+q_1q_2) & q_1^2-q_2^2-q_3^2+q_4^2 \end{pmatrix},$$

where $|q|^2 = q_1^2+q_2^2+q_3^2+q_4^2$. A projection matrix P can be constructed from X and the principal point (p_x, p_y) as follows:

$$P = \begin{pmatrix} \phi & 0 & p_x & 0 \\ 0 & \phi & p_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} & & & 0 \\ & R & & 0 \\ & & & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

This calibration determines the alignment of the camera with respect to a world coordinate frame. It fixes an arbitrary value for the translation of the camera centre from the world origin. There remains an unknown scaling to convert between pixel units and a world metric which could be computed given a manual measurement of the real-world length of any line in the image which is parallel to one of the world coordinate axes.

2.2 Manhattan likelihood

Coughlan and Yuille [5] describe a likelihood function for a Manhattan world image Z given a calibration X which we use unaltered, but for completeness it is described in this section. The likelihood decomposes into a product of independent likelihoods

$$p(Z|X) = \prod_{k=1}^{K} p_{\mathbf{x}_k}(Z|X)$$

where $\mathbf{x}_k$ is an image coordinate. The $\mathbf{x}_k$ may be sampled densely at each image pixel, or sparsely on a grid. Sparse sampling has the advantages of reducing processing time and also of mitigating the effects of the (incorrect) assumption that the per-pixel likelihoods are independent. We use a sparse grid with pixel spacing γ. Rather than holding the grid fixed, for example by using the pixel at the centre of each $\gamma \times \gamma$ grid square, we got better results by choosing a pixel at random from each grid square. We perform this randomisation within the grid independently *each time the likelihood is evaluated* which has the effect of averaging the image data rather than only using a $1/\gamma^2$ fraction of it.

The Manhattan assumption states that we expect to see many edges aligned with one of the three world-coordinate axes, so the per-pixel likelihood further decomposes into a mixture over five models m_i

$$p_{\mathbf{x}_k}(Z|X) = \sum_{i=1}^{5} p_{\mathbf{x}_k}(Z|m_i, X)P(m_i)$$

where models m_1, m_2, m_3 correspond to the event that an edge in one of the coordinate directions is imaged at point $\mathbf{x}_k$, m_4 means that an edge in a random direction is imaged at that point and m_5 means that no edge is imaged at $\mathbf{x}_k$. Following [5] we adopt fixed prior probabilities on the five models:

$$P(m_i) = \{0.02, 0.02, 0.02, 0.04, 0.90\}.$$

The pixel likelihood for each model is given by

$$p_{\mathbf{x}_k}(Z|m_i, X) = \begin{cases} p^{\text{on}}_{\mathbf{x}_k}(Z)p^{\text{ang}}_{\mathbf{x}_k}(Z|\theta^i_{\mathbf{x}_k}(X)), & i = 1, 2, 3 \\ p^{\text{on}}_{\mathbf{x}_k}(Z)\frac{1}{2\pi}, & i = 4 \\ p^{\text{off}}_{\mathbf{x}_k}(Z)\frac{1}{2\pi}, & i = 5 \end{cases}$$

where $p^{\text{on}}_{\mathbf{x}_k}(Z)$ is the probability that an edge in the world was imaged to generate the pixel values in the neighbourhood of $\mathbf{x}_k$ and $p^{\text{off}}_{\mathbf{x}_k}(Z)$ is the probability that a non-edge region generated the neighbourhood of $\mathbf{x}_k$. Both p^{on} and p^{off} are learnt functions of the gradient magnitude at $\mathbf{x}_k$ and details can be found in [5] — the same learnt functions are used here as in that paper. The angle $\theta^i_{\mathbf{x}_k}(X)$ specifies the direction in which coordinate axis i will be imaged at $\mathbf{x}_k$ according to the hypothesised calibration X and $p^{\text{ang}}_{\mathbf{x}_k}(Z|\theta)$ specifies the probability of observing

the pixels in the neighbourhood of $\mathbf{x}_k$ given a world edge which should image in direction θ. Again following [5] we set

$$p^{\text{ang}}_{\mathbf{x}_k}(Z|\theta) = \begin{cases} p^{\theta}(\alpha(\mathbf{x}_k) - \theta) \text{ if } |\alpha(\mathbf{x}_k) - \theta| < \theta_0 \\ p^{\theta_0} \qquad\qquad\qquad \text{otherwise} \end{cases}$$

where $\alpha(\mathbf{x})$ gives the measured gradient direction at an image point $\mathbf{x}$ and $p^{\theta}(\lambda)$ gives the probability of observing an edge deviating by an angle λ from the predicted direction; p^{θ} is taken to be a triangular distribution with width θ_0, where p^{θ} and p^{θ_0} are normalised so that the pdf is continuous and

$$\int_{\theta=0}^{2\pi} p^{\text{ang}}_{\mathbf{x}_k}(Z|\theta) d\theta = 1.$$

2.3 Density estimation

The stochastic search algorithm we adopt is a form of iterated importance sampling using kernel density estimates as the importance functions [13]. We aim to estimate the posterior density $f(X) \equiv p(Z|X)p(X)$ with a set of N weighted particles (s_n, π_n); each s_n is a choice of calibration parameters, and each π_n is a real-valued weight. The density estimated by the particle set (s_n, π_n), $n = 1, \ldots, N$ can be thought of as just the weighted sum $\sum_{n=1}^{N} \pi_n \delta_{s_n}$ of δ-functions centred at the particle locations. (Of course the weights should be rescaled so that they sum to 1 before the density is computed.) *Importance sampling* is an algorithm for producing such particle sets. Given a density $g(X)$ which is to be used as an importance function, the particle set is generated by randomly sampling from g for $n = 1$ to N as follows:

$$s_n \sim g(X), \quad \pi_n = \frac{f(s_n)}{g(s_n)},$$

and this is well known to be an increasingly efficient representation of the posterior as g is increasingly similar to f. We therefore adopt an iterated scheme, performing importance sampling K times with importance functions g^k where $g^1(X) \equiv p(X)$ is a prior over X, and g^k for $k > 1$ is formed as a kernel density based on the particle set resulting from step $k-1$. Let S^k be the particle set (s^k_n, π^k_n) generated by importance sampling using g^k at step k, then we can construct a mixture density g^{k+1} from S^k as follows:

$$g^{k+1} = \sum_{n=1}^{N} \pi^k_n \mathcal{K}_{s^k_n}, \tag{1}$$

where $\mathcal{K}_{s^k_n}$ is a suitable kernel centred at s^k_n.

Since the likelihood is sharply peaked near the true calibration, we adopt an annealing approach, so that at iteration k rather than using the posterior f we instead use a distribution with smoothed likelihood

$$f^k(X) = (p(Z|X))^{\beta^k} p(X)$$

where $\beta^k \leq 1$ is an annealing parameter and $\beta^K = 1$ so the final estimate approximates the desired posterior. We also use a sequence of kernels $\mathcal{K}^k$ which decrease in size (i.e. covariance) as k increases, so the state-space is explored more widely at early iterations of the algorithm. We vary the number of particles N^k to use more particles in early iterations.

The kernels $\mathcal{K}^k$, annealing parameters β^k, number N^k of particles and number K of iterations are set by hand: this is known to be an art [13] but we have not found the algorithm to be particularly sensitive to the choice of these parameters. In section 3 we run the algorithm multiple times for each example image using different random seeds to verify that the estimates produced are consistent.

We use as a prior on X a uniform distribution over the space of rotations and a uniform distribution within a specified interval for the focal length

$$\phi \sim \mathcal{U}[\phi_{\min}, \phi_{\max}].$$

The uniformly distributed rotation is generated by independently sampling each $q_i, i = 1 \ldots 4$ from a standard normal distribution. The focal length component of kernel $\mathcal{K}_x^k$ is taken to be the Gaussian $\mathcal{N}(\phi, \sigma_\phi^k)$. Placing a kernel in rotation space is slightly trickier, but we use the following approximation which is valid for small perturbations in rotation space. To sample from the kernel $\mathcal{K}_u(u')$ about a rotation u we generate a perturbation rotation v

$$v = (1, v_2, v_3, v_4), \quad v_i \sim \mathcal{N}(0, \sigma_v)$$

and set the perturbed rotation u' to be the quaternion product $u' = uv$. To use equation (1) we also need a way of evaluating $\mathcal{K}_u(u')$. This is done by setting $v' = (u')^{-1}u$ and writing

$$\bar{v}' = (1, v_2'/v_1', v_3'/v_1', v_4'/v_1').$$

We can then approximate

$$\mathcal{K}_u(u') = \prod_{j=2}^{4} \frac{1}{\sqrt{2\pi}\sigma_v} \exp(-\frac{(\bar{v}_j')^2}{2\sigma_v^2}).$$

Having estimated the posterior distribution as a particle set S^K we would like to output a single parameter vector $\hat{X}$ as the calibration estimate. We expect that the posterior will have many local maxima but be dominated by a peak at the correct calibration parameterisation. Since it is difficult to estimate the mode of a particle set, we instead assume that the posterior will be sufficiently peaked that the mean of the distribution is a good approximation to the mode. This mean is given by

$$\hat{X} = \sum_{n=1}^{N} \pi_n^K s_n^K.$$

This weighted mean suffices to estimate the focal length parameter ϕ but poses some problems for the quaternion rotation. Recall that X is being estimated to

line up with the three world coordinate axes. Since the axes are not labelled, there is a 24-fold symmetry: three choices for which axis is vertical, then two choices for which way is up, then four choices for which way is North. The analogous problem in one dimension would be to take a weighted sum of angled lines where the direction of the line α is not important so α is identified with $\pi+\alpha$ (a 2-fold symmetry). The solution we adopt is to round each quaternion to a canonical value in $SO(3)/C$ where C is the octohedral group of the symmetries of a cube. The rounding algorithm is given in figure 1 but space does not permit us to include its derivation which can be found in [15]. In the one-dimensional case this would correspond to adding and subtracting multiples of π to ensure that each angle lies within the range $0 \leq \alpha < \pi$. Note that this has not entirely solved our problem if the particles lie near to the boundary of the rounding cell: imagine that the angles are clustered around $[0, \delta) \cup (\pi - \delta, \pi)$. Although there are solutions to this problem in the simple case of angled lines, we do not know of a clean solution in the case of $SO(3)/C$ so we currently proceed in the hope that the problem will not frequently arise.

To round a quaternion $u = (a, b, c, d)$ into a canonical value u' in $SO(3)/C$:

1. Set (p, q, r, s) to be the result of sorting the sequence $(|a|, |b|, |c|, |d|)$ into nonincreasing order, so
$$p = \max(|a|, |b|, |c|, |d|)$$
$$s = \min(|a|, |b|, |c|, |d|).$$
2. Find the rotation $v \in C$ closest to u. First compute the following three values, and test which is largest: $p, (p+q)/\sqrt{(2)}$ and $(p+q+r+s)/2$.
   (a) if p is largest, determine which of the four numbers $|a|, |b|, |c|, |d|$ is largest (and is hence equal to p), then convert u into v by replacing that number with +1 or -1 according to its sign and the other three numbers with 0.
   (b) if $(p+q)/\sqrt{(2)}$ is largest, determine which two of the four numbers $|a|, |b|, |c|, |d|$ are largest (and hence equal to p and q), then convert u into v by replacing those numbers with +1 or -1 according to their signs and the other two numbers with 0.
   (c) if $(p+q+r+s)/2$ is largest, convert u into v by replacing all four numbers with +1 or -1 according to their signs.
3. Return $u' = uv^{-1}$.

Fig. 1. Rounding a quaternion into $SO(3)/C$.

3 Results

We performed experiments using three different digital cameras, and chose two focal-length settings for each camera giving six distinct internal calibrations in

all. For each calibration we took multiple pictures of a planar calibration grid and estimated ground-truth focal length using the code supplied by [2]. We then took a number of images in our office building with each camera to test our algorithm. Some of the images could be termed "poorly-conditioned" for our algorithm. There are at least two types of such poorly-conditioned images. In the first type, the vanishing points cannot be easily inferred due to the lack of oriented edges or textures in the image. An obvious example would be an image in which only a single plane is visible. In the second type of poorly-conditioned image, two of the three vanishing points corresponding to orthogonal directions are at infinity. It is well-known that focal length estimation is singular in this case [3, 9]. A camera pointing directly down a corridor, and parallel to the ground, is an example of this second type of poorly-conditioned image. We made no particular effort to avoid poorly-conditioned images in the test set, and have included results on all of the images we took to demonstrate both successes and failures of the algorithm.

The same parameter settings were used in all experiments as follows: the likelihood parameters were a grid spacing $\gamma = 5$ and $\theta_0 = 10$ degrees, and the focal length prior was taken to be uniform over $[100, 2500]$ pixels. The algorithm was run for $K = 40$ iterations with parameters given in table 1. The algorithm

k	1–10	11–20	21–30	31–40
N^k	2500	1500	1500	1500
σ_v^k	0.0146	0.0104	0.0074	0.0052
σ_ϕ^k	36.50	26.00	18.50	13.00
β^k	0.125	0.250	0.500	1.000

Table 1. Parameter settings for iterations $k = 1$ to K.

was run ten times with different random seeds for each test image. The estimated focal lengths are shown in figure 2 along with the ground truth value computed separately. Reprojected results for one of the test runs on each image are shown for cameras 1–3 with settings A and B in figures 3–8. White lines are shown pointing towards the Manhattan vanishing points of the images. The numbering of the images in figure 2 corresponds to raster order in figures 3–8.

For all cameras except $3A$ the estimated focal lengths cluster around the ground truth. Note that when the algorithm correctly finds the focal length multiple runs with different random seeds produce similar estimates, with the exception of a single outlier for camera $2A$ image 3. The algorithm performs poorly on several of the test images for camera $3A$. As can be seen in figure 7, images 7 and 8 do not contain a significant number of edges in one of the three world-axes, so it is unsurprising that the calibration is poorly estimated. It is less apparent why images 1 and 4 pose problems: probably image 1 is too dark and the edges in the direction pointing away down the corridor in image 4 are

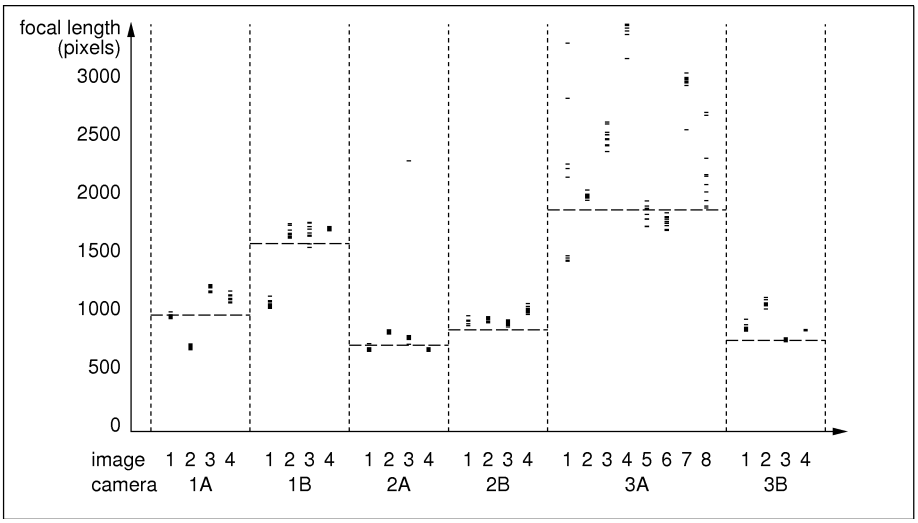

Fig. 2. Estimated focal lengths. The algorithm was run on several trial images for each of six camera/focal length settings. For each camera, the ground-truth focal length is shown as a dashed horizontal line. The algorithm was run ten times on each test image and the ten estimates are shown here as short horizontal lines. The results are analysed in section 3.

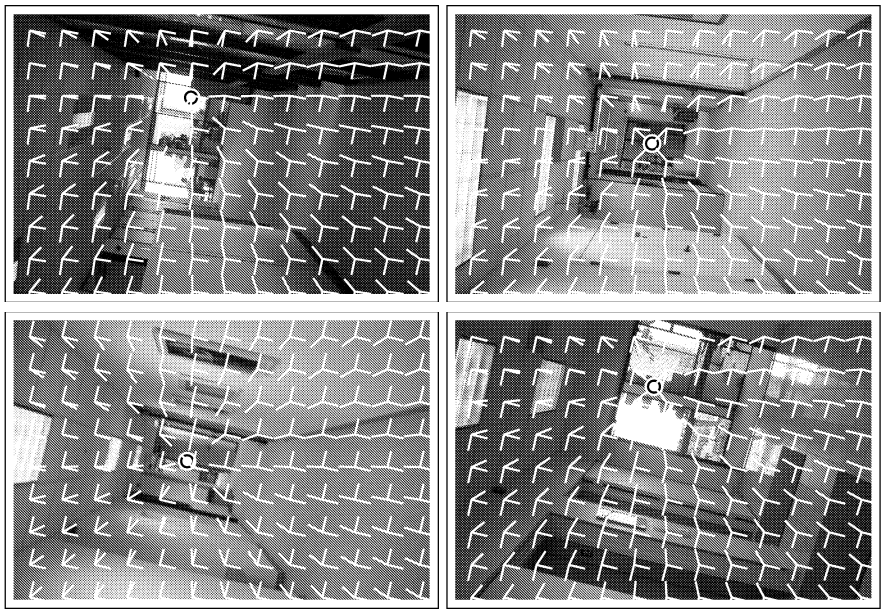

Fig. 3. Test calibrations for camera $1A$**.** White lines point towards the vanishing points (shown as a bullseye where visible) of the estimated world axes.

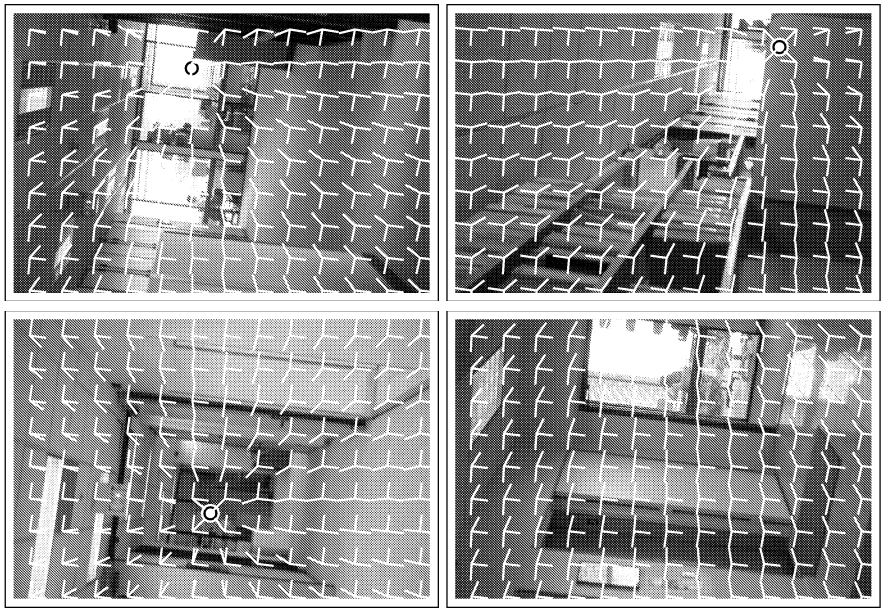

Fig. 4. Test calibrations for camera $1B$**.** White lines point towards the vanishing points (shown as a bullseye where visible) of the estimated world axes.

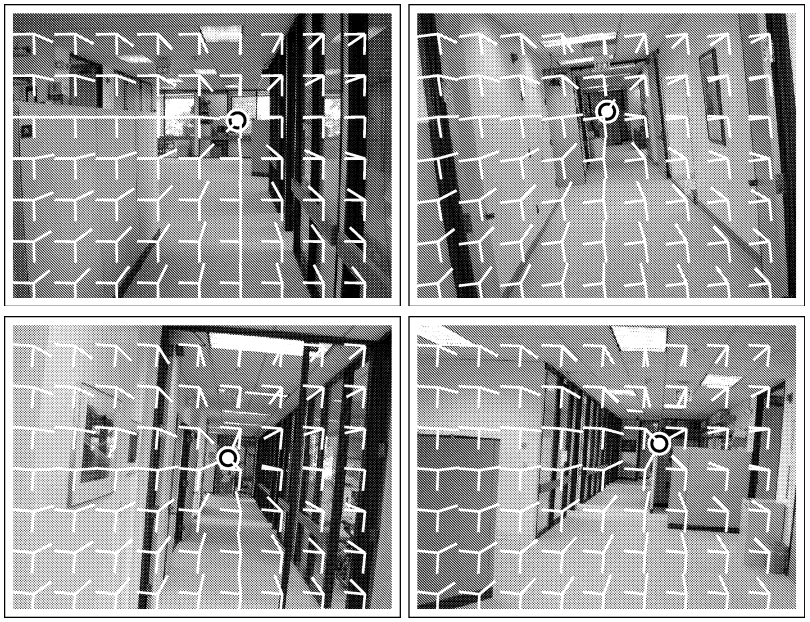

Fig. 5. Test calibrations for camera $2A$**.** White lines point towards the vanishing points (shown as a bullseye where visible) of the estimated world axes.

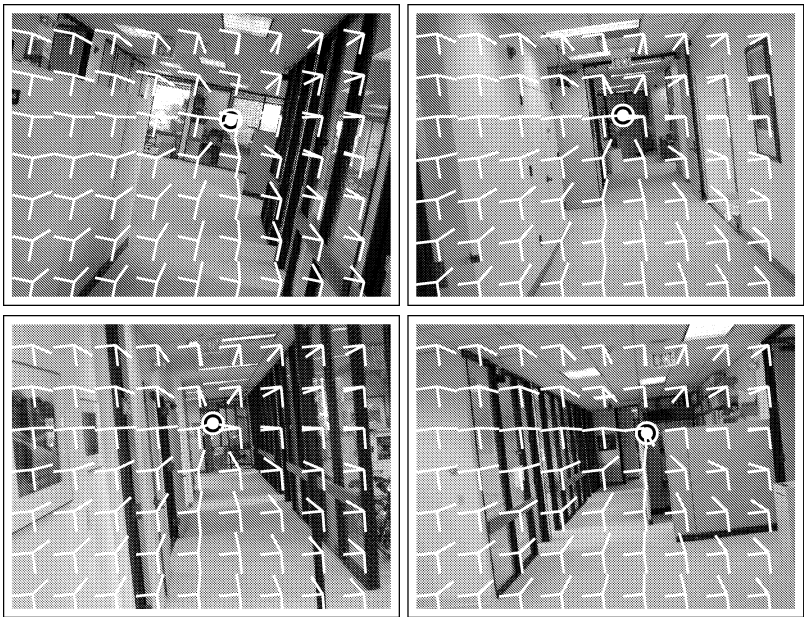

Fig. 6. Test calibrations for camera $2B$**.** White lines point towards the vanishing points (shown as a bullseye where visible) of the estimated world axes.

too faint. In addition, image 4 is close to the singular configuration referred to above, in which two of the vanishing points are at infinity.

4 Conclusion

We demonstrate that a single image of an office environment is sufficient to automatically calibrate a camera with reasonable accuracy, given some weak assumptions about the nature of the scene statistics and the likely internal camera parameters. The algorithm can be applied without user intervention, using fixed parameter settings, and works on a wide variety of images and camera geometries. The method is of practical use for surveillance applications — indeed we currently use it to provide calibrations for a person-tracking system which has been presented elsewhere [10].

It may be interesting to extend the Manhattan calibration algorithm to more general scenes. We have assumed that there are three principal directions present in the scene and that they are orthogonal, however all of the likelihood calculations would go through unchanged given an arbitrary number of directions in arbritrary configuration. It might be possible to refine the search process to estimate an unknown number of principal directions in a scene, to cope for example with buildings whose rooms are not all rectangular.

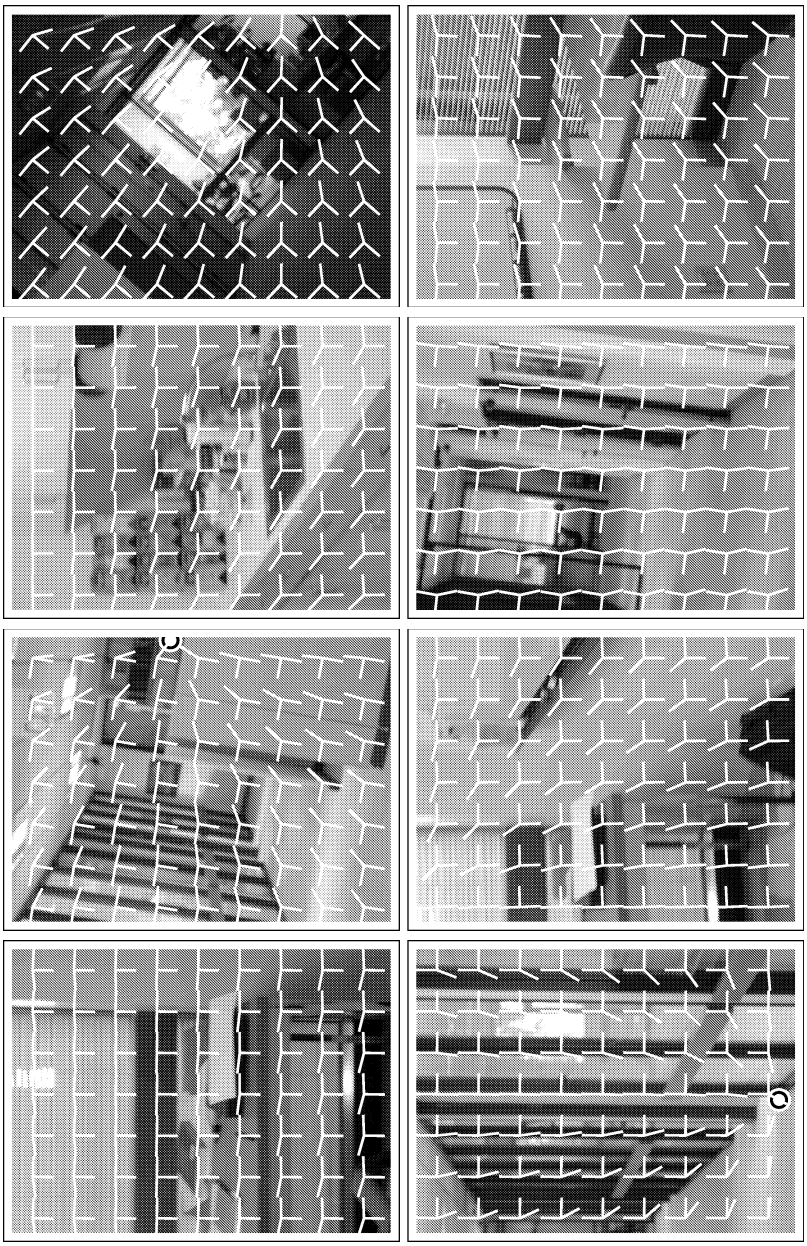

Fig. 7. Test calibrations for camera $3A$**.** White lines point towards the vanishing points (shown as a bullseye where visible) of the estimated world axes.

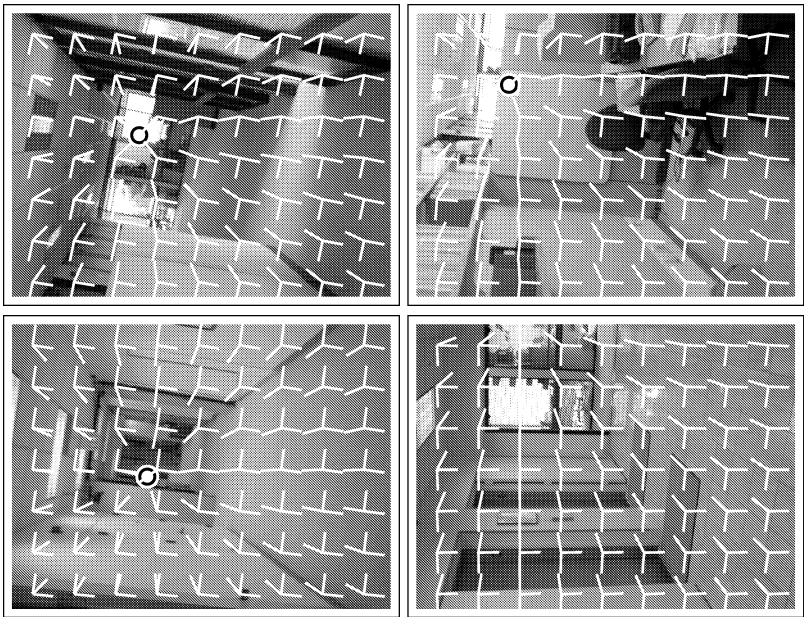

Fig. 8. Test calibrations for camera $3B$**.** White lines point towards the vanishing points (shown as a bullseye where visible) of the estimated world axes.

We hope that these results will encourage interest within the field in applying statistics to geometry. In particular, as here, statistical assumptions may be useful in regularising geometric vision problems which are widely considered insoluble when the available data do not determine a unique solution.

Acknowledgments

The authors would like to thank Lyle Ramshaw for many helpful suggestions and discussions, and especially for the method of canonicalising rotation matrices. Alan Yuille provided raw data for the learnt pixel likelihoods. The anonymous referees and the area chair provided helpful comments.

References

1. M. Antone and S. Teller. Automatic recovery of relative camera rotations for urban scenes. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 2, pages 282–289, 2000.
2. J-Y. Bouguet. Camera calibration for Matlab. Technical report, Intel Corporation, available from `http://www.vision.caltech.edu/bouguetj/calib_doc/`, 2001.
3. B. Caprile and V. Torre. Using vanishing points for camera calibration. *Int. J. Computer Vision*, 4(2):127–140, 1990.

4. R. Cipolla, T. Drummond, and D. Robertson. Camera calibration from vanishing points in images of architectural scenes. In *Proc. British Machine Vision Conference*, pages 382–391, 1999.
5. J.M. Coughlan and A.L. Yuille. Manhattan world: Compass direction from a single image by Bayesian inference. In *Proc. 7th Int. Conf. on Computer Vision*, pages 941–947, 1999.
6. A. Criminisi, I.D. Reid, and A. Zisserman. Single view metrology. *Int. J. Computer Vision*, 40(2):123–148, 2000.
7. K. Daniilidis and J. Ernst. Active intrinsic calibration using vanishing points. *Pattern Recognition Letters*, 17(11):1179–1189, 1996.
8. O. Faugeras. *3D Computer Vision*. MIT Press, 1993.
9. R. Hartley and A. Zisserman. *Multiple View Geometry in computer vision*. Cambridge University Press, 2000.
10. M. Isard and J. MacCormick. BraMBLe: A Bayesian multiple-blob tracker. In *Proc. 8th Int. Conf. Computer Vision*, volume 2, pages 34–41, 2001.
11. K. Kanatani. Statistical analysis of focal-length calibration using vanishing points. *IEEE Trans. Robotics and Automation*, 8(6), 1992.
12. D. Liebowitz and A. Zisserman. Metric rectification for perspective images of planes. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 482–488, 1998.
13. G. MacLachlan and K. Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, 1988.
14. G. McLean and D. Kotturi. Vanishing point detection by line clustering. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(11):1090–1094, 1995.
15. L. Ramshaw. Averaging on a quotient of a sphere, 2002. Technical note 2002-003, Compaq Systems Research Center, Palo Alto, CA.
16. R.Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE J. of Robotics and Automation*, 3(4):323–344, 1987.
17. A. Watt and M. Watt. *Advanced Animation and Rendering Techniques*. Addison-Wesley, 1992.
18. Z. Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.