

Unsupervised model-free camera calibration algorithm for robotic applications

Guglielmo Montone, J. Kevin O'Regan and Alexander V. Terekhov

Abstract—This paper presents an algorithm for camera calibration. The algorithm, inspired by work in the field of developmental robotic on the concept of space in naïve agents, is particularly suitable for robotic applications: it is completely unsupervised, and it does not assume any model of the camera, making it applicable to many kinds of optical devices. Testing of the algorithm, in a simulated environment, shows very good results, outperforming the main unsupervised and model-free calibration algorithm in the literature.

I. INTRODUCTION

Mobile robots are frequently faced with the problem of operating in an uncertain and unpredictable environment using complex sensory systems, such as camera, laser, and sonar. Proper calibration of such sensors is a necessary condition to perform many tasks. Examples include object recognition as well as many tasks involved in navigation, such as evaluating distances among landmarks and measuring robot rigid displacements. Indeed, the problem of sensor calibration is strongly related to the problem of equipping a naïve agent with algorithms to perform space-related measurements. Let us consider the problem of calibrating a generic visual system composed of different lenses and a plane surface over which photoreceptors are placed according to some arbitrary distribution. Calibrating this system consists in defining an algorithm able to associate each photoreceptor to its relative direction of sight in the external world. Equivalently, a robot equipped with a calibrated visual system would be able to perform distance measurements, which in this case would consist in angular distances in the camera's field of view (FOV).

In recent years a variety of robotic systems have been introduced which use vision sensors to perform navigation and object recognition tasks. In these systems the calibration of the camera is often performed before the camera is installed on the robot. This is because of some constraints that the calibration procedure needs to fulfill: human supervision [1] and the use of a calibration pattern [2] are two such types of restrictions. Moreover, many calibration techniques make use of models of both the camera and the distortions induced by the lenses [3], [4], [5], limiting their application just to systems that are described by these models.

These restrictions are in contrast with one of the important aims of current robotics research: to create robots which are as autonomous as possible. A calibration algorithm for these

robots should be: 1) completely autonomous: the calibration should be performed by means of a completely unsupervised algorithm; 2) as general as possible: the calibration algorithm should be applicable to a wide range of optical devices and 3) able to work in any kind of environment. Few techniques ([6], [7], [8]) with these characteristics have been proposed until now. These techniques are mainly inspired Kuipers' [9] work in developmental robotic. Kuipers' calibration algorithm estimates the distances among the directions of sight of photoreceptors by evaluating the similarity among the input values of these photoreceptors. The implied assumption in Kuipers' algorithm is the following: for a given pair of areas in the camera's FOV, the greater the distance between them, the greater the difference in the corresponding sensory input to the camera. This assumption, although locally true, is not a global property of space, implying that Kuipers' algorithm will only work properly when the distances to be evaluated are small compared to the speed of change in the image in the camera's FOV.

In this paper we propose a new method for camera calibration that fulfils the three requirements listed above. This method is inspired by work in developmental robotic on the emergence of the concept of space in naïve agents [10], [11]. In particular, two works by Terekhov and O'Regan [12], [13] show how a naïve robot equipped with a simple visual system can be able to measure what physicists commonly call rigid displacements. The key characteristics of Terekhov and O'Regan's algorithm and its main difference from that of Kuipers, is the importance given to camera movements. In their algorithm, camera movements are fundamental to how the agent defines rigid displacements, in the sense that the agent can define and recognize rigid displacements in its own coordinates system (here, the set of photoreceptors of the camera) once it notices that the input sensed by a given photoreceptor i before a specific displacement d will always be sensed by another photoreceptor j after the displacement. This characteristic is a distinguishing property of space and of rigid displacements in the sense that it is completely independent of the environment and the particular characteristics of the camera. The main contribution of this paper consists in showing how this property can be exploited to implement a camera calibration algorithm, and demonstrating the advantages of the proposed calibration compared to that of Kuipers.

The rest of the paper is organized in the following way. In section II the proposed calibration algorithm is explained in detail together with a description of our implementation of Kuipers' method. Both techniques were tested by calibrating

*This work was funded by the European Research Council (FP 7 Program) ERC Advanced Grant "FEEL" to KO'R.

The authors are with Laboratoire Psychologie de la Perception, Université Paris Descartes, 75006 Paris, France. guglielmo.montone, jkevin.oregan, avterekhov@gmail.com

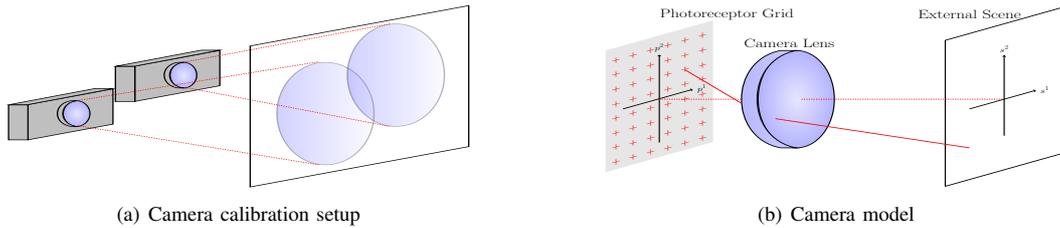


Fig. 1. (a) Camera calibration setup. The camera is calibrated by taking pictures of planar images presented on a plane parallel to the plane of the camera photoreceptors. The camera movements considered are translations in this plane. The dotted red lines delimit the camera’s field of view (FOV). In the case shown here the two FOVs overlap. (b) Camera model. The grey plane on the left represents the camera’s photoreceptor plane. The photoreceptors are indicated by red crosses. The blue cylinder in the center of the figure represents the camera’s lens. The red dotted line is the camera’s optical axis. The transparent plane on the right is the one over which the pixels’ lines of sight are projected. The red solid line represents the direction of sight of a particular photoreceptor, which is deflected by the lens.

a pin-hole camera and a fish-eye camera in a simulated environment. Details on the simulated experiments and the results of the two calibrations are presented in section III.

II. DETAILS ON THE ALGORITHM

Before formalizing the problem of camera calibration, let us clarify the robotic setting we are considering. Let us imagine a naïve agent scanning a planar image with a camera and assume that the robot is able to perform rigid displacement of the camera in front of the observed image (see figure 1(a)). Now let us consider the robot to be completely unaware of the camera model, in the sense that the agent does not know anything about the lenses of the camera and how they distort the image of the world. Moreover, let us assume that the agent is able to measure the displacements of the camera using an odometry sensor which is quite noisy. Clearly the presence of the distorting lenses means that the image captured by the camera is not an isometric representation of space. The distance between the projections of two luminous spots may therefore vary.

Let us consider what happens when the camera performs a small rigid displacement such that the FOVs of the camera before and after the movement overlap. In this condition, the input that was sensed by some photoreceptor before the displacement would be sensed by some other photoreceptor after the displacement. The agent could store these coincidences by defining a map, in the photoreceptor set, coupling photoreceptors that sensed the same input before and after the displacement. We will call such maps φ functions: one such map could be univocally associated to each of the possible rigid displacements of the camera in the world. In fact it will be always the case that different movements will be associated to different φ maps. Moreover, for a given movement of the camera the agent will discover the same φ function, even if the displacement starts from a different initial position and for a different observed image. In the next paragraph we will describe the algorithm for evaluating the φ functions and how these function can be used to estimate the distances among photoreceptors.

Once the distances among the photoreceptors’ directions of sight have been estimated, the problem of finding these directions can be considered a problem of metric embedding,

frequently referred as multidimensional scaling [14]. In the following paragraph we will quickly state the problem of camera calibration in a more formal fashion; this is followed by detailed presentation of our implementation of Kuipers’ algorithm and our own technique.

A. Camera Calibration

The problem of camera calibration can be stated in the following way. Referring to figure 1(b), let us call $\mathcal{P} = \{p_i\}_{i=1}^n$ the set of two-dimensional vectors whose components are the coordinates of the photoreceptors in a Cartesian coordinate system fixed to the camera’s photoreceptor plane. Considering a plane facing the camera, orthogonal to the camera’s optical axis, and providing it with a Cartesian coordinate system, we can define $\mathcal{S} = \{s_i\}_{i=1}^n$ as the set of directions of sight for each photoreceptor. Each s_i is a two-dimensional vector whose components are the coordinates of the intersection between the i -th photoreceptor’s direction of sight and the plane facing the camera (see figure 1(b)). Finally let us call $\rho(s_i, s_j)$ the Euclidean distance between the directions of sight of the i -th and j -th photoreceptor evaluated on the plane where s_i and s_j are defined.

The *camera calibration problem* consists in estimating, for each photoreceptor p_i , its relative direction of sight s_i . One general solution for this problem consists in: 1) defining a dissimilarity function $\mu_{ij} = \mu(p_i, p_j)$ on the set \mathcal{P} , 2) applying a multidimensional scaling algorithm which returns the set \mathcal{S} as the result of the following minimization:

$$\tilde{\mathcal{F}} = \min_{f, s'_1, \dots, s'_n} \frac{\sum_{ij} \left(\rho(s'_i, s'_j) - f(\mu_{ij}) \right)^2}{\sum_{ij} \rho(s'_i, s'_j)^2}; \quad (1)$$

where f is a monotonic functions. Depending on how μ_{ij} is determined, different approaches result.

In the two following paragraphs we will describe how the dissimilarity μ is defined and computed in both Kuipers’ algorithm and ours. In describing the two algorithms, we will refer to the experimental setting represented in figure 1(a). The camera is considered to be set in front of an image which lies on a plane perpendicular to the camera’s optical axis. References to camera movements will be to translations of the camera in the camera’s photoreceptors plane, i.e. in a

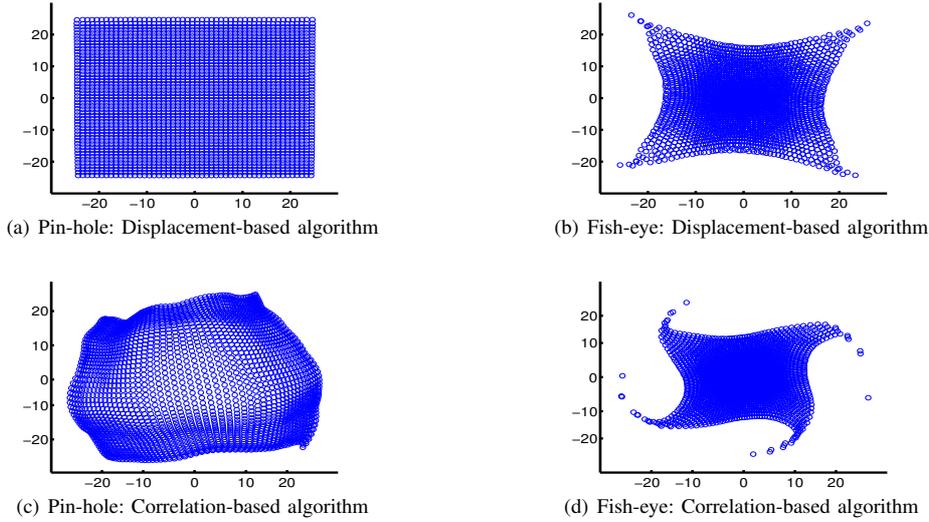


Fig. 2. The coordinates of the estimated directions of sight \tilde{s} for each photoreceptor in the square photoreceptor grid (size 50) of a pin-hole camera((a) and (c)) and a fish-eye camera((b) and (d)). The top row presents the results of our calibration methods with 10% noise in movement estimation, the bottom row shows the results of Kuipers’s algorithm. The coordinates \tilde{s} were rescaled in order to yield a maximum value of $\rho(\tilde{s}_i, \tilde{s}_j)$ of approximately 50, and the points were rotated in such a way the side of the estimated grid approximately parallel to the reference system.

plane parallel to the image. More details on the images used for the calibration are given in the *Implementation details* section.

B. Correlation-based algorithm

Here we have named the calibration technique obtained using Kuipers’ definition of dissimilarity μ the *correlation-based* algorithm. Our main reference in implementing Kuipers’ algorithm was [15]. Defining $X(t) = \{x_i(t)\}_{i=1}^n$ as the values sensed by all photoreceptors at time t , Kuipers’ algorithm consists in collecting different $X(t)$ at different times while moving the camera around. In the setup described in 1(a), we collected a series of vectors $[X(1), \dots, X(T)]$, by performing T random translations of the camera in the photoreceptor plane. The dissimilarity μ (from now on μ^C , with the superscript C standing for correlation-based) between two photoreceptors was then estimated as:

$$\mu_{ij}^C = \mu^C(p_i, p_j) \stackrel{\text{def}}{=} \frac{1}{T} \sum_{t=1}^T |x_i(t) - x_j(t)|. \quad (2)$$

Here the distance between the directions of sight of two photoreceptors is estimated by considering the mean difference over time in the input sensed by the photoreceptors. The underlying assumption is that two photoreceptors p_i and p_j , with similar directions of sight s_i and s_j , sense approximately the same input. The greater the distance between s_i and s_j , the more different will be the inputs received by the photoreceptors. Clearly this assumption holds if the observed image is smooth enough, or, more precisely, if the distances among the directions of sight to be estimated are small compared with the speed of changing of the observed images. When the distances become larger than this, the estimate μ^C reaches a constant value and becomes distance independent.

C. Displacement-based algorithm

In this paragraph we will propose our definition of dissimilarity μ - hereinafter μ^D - describe the algorithm to evaluate it. We will refer to this algorithm as the *displacement-based* calibration algorithm. Let us define \mathcal{D} as the set of all rigid displacements the camera can perform such that at least a portion of the FOV of the camera’s FOV before the displacement is still in the camera’s FOV after the displacement. Let x_i be the input sensed by the i -th photoreceptor before performing the displacement d , and let y_i be the input sensed by the same photoreceptor after performing the displacement d . We collect the set of data $X_i^d = \{x_i^k\}_{k=1}^M$ and $Y_i^d = \{y_i^k\}_{k=1}^M$, for all of the camera’s pixels by performing M times the displacement d . We expect that the closer the direction of sight s_i of the i -th photoreceptor before the displacement is to the direction of sight s_j of the j -th photoreceptor after the displacement, the more similar the sequences X_i^d and Y_j^d will be. We associated a bijective function $\varphi_d: \mathcal{P} \rightarrow \mathcal{P}$ on the photoreceptors set to each displacements $d \in \mathcal{D}$ as follows:

$$\varphi_d(p_i) = \arg \max_{p_j \in \mathcal{P}} |corr(X_i, Y_j)|; \quad (3)$$

where $corr(X_i, Y_j)$ is the correlation between the two sequences of photoreceptor values.

Let us now see how we can use φ to estimate the distances between the directions of sight s_i, s_j of two photoreceptors. The φ_d function couples together photoreceptors whose respective directions of sight s happen to coincide before and after a the displacement d . For this reason we would like $\mu^D(p_i, p_j)$ to be the same for all the pairs of photoreceptors that are associated by the same φ function. The dissimilarity μ^D we propose is the following:

$$\mu_{ij}^D = \mu^D(p_i, p_j) \stackrel{\text{def}}{=} \|\varphi_d\| \quad \text{where} \quad \varphi_d(p_i) = p_j. \quad (4)$$

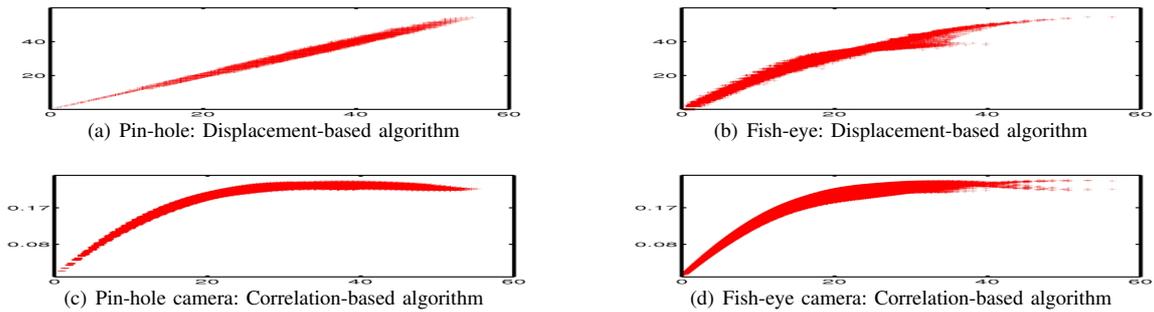


Fig. 3. The dissimilarities μ_{ij}^D and μ_{ij}^C versus the distance $\rho(s_i, s_j)$ for a pin-hole camera and a fish-eye camera with a photoreceptor grid of size 40. The results on the displacement-based algorithm refer to the calibration condition where the displacements were corrupted with Gaussian error.

The norm used $\|\cdot\|$ is such that the identity function has a zero norm. We decided to use the following norm here:

$$\|\varphi_d\| = \arg \max_{p_j \in \mathcal{P}} \|\varphi_d(p_j) - p_j\|. \quad (5)$$

The dissimilarity function μ^D has some interesting property. It is symmetric, as far as, we define, for each φ_d function for the displacement d , a φ_{-d} for the opposite displacement. Moreover, if we build the set of φ functions in such a way that, for any three photoreceptors in \mathcal{P} there is always one or more φ functions which maps any of the three photoreceptors onto the other, then the triangular inequality will hold for μ^D . We thus have that, if the set of φ functions is properly built, then μ^D is actually a metric on \mathcal{P} .

III. EXPERIMENTS AND RESULTS

A. Implementation details

The displacement-based and correlation-based algorithms were both tested by calibrating two kinds of cameras: a pin-hole camera and a fish-eye camera. The camera calibration setup is described in the previous section and is depicted in figure 1(a). The calibration algorithms were tested on a camera equipped with a square grid of photoreceptors of different sizes, with the number of photoreceptors per grid side N ranging from 20 to 70.

The images used for the camera calibration were grayscale images composed of random patches. The smoothness of the images used was varied during the calibration within a range compatible with that of an off-the-shelf camera¹.

As stated in the previous paragraph, it is possible to associate a φ function to each camera movement, where the camera's FOV before and after the displacement partially overlap. The displacements used for the calibration were translations of the camera in the photoreceptor plane. Considering that the size of the grid of photoreceptors is $L = N \cdot l$, where N is the number of photoreceptors per size and l is the step of the grid, a set $d' = \{-L, -L+l, -L+2l, \dots, L-2l, L-l, L\}$ can be defined. The set of displacements used for the calibration

¹The smoothness of a gray-level image taken with a off-the-shelf camera can vary widely depending on the resolution of the camera and the mean distance to the objects in the scene, but, the mean absolute derivative for these images usually ranges in the interval $I = [0.01, 0.05]$.

was $D = d' \otimes d'$, where the vectors in the set D are defined in relationship to a reference frame with origin at the center of the photoreceptors grid and axes parallel the sides of the grid (see figure 1(b)). In the correlation-based algorithm inspired by Kuipers, in order to collect the inputs to the photoreceptors to evaluate distances according to eq. 2, we moved the camera through 5000 displacements randomly drawn from the set D . Every 250 movements a new image was shown to the camera.

In the displacement-based method a φ function was associated to each of the displacements D according to equation 3. In order to evaluate the correlation among photoreceptor values, the same camera displacement was repeated 20 times. Each time the movement was repeated a new random image was shown to the camera. In displacement-based calibration the camera must be moved multiple times through the same displacement. This could be a quite restricting condition in a real robotic setting, where the robot displacement is affected by error. We thus chose to simulate the camera calibration in two different conditions. In the first setting the camera displacements were not affected by any error, while in the second setting a Gaussian error of 10% was applied to the norm of the displacement.

B. Results

Once the dissimilarities, μ^C and μ^D , among photoreceptors have been evaluated according to the equations 2 and 4, we applied multidimensional scaling (STRESS 1) to obtain for each photoreceptor p_i the estimated coordinates \tilde{s}_i . In displaying the results we sometimes multiplied the estimated direction of sights by a constant factor α or isometrically transformed the \tilde{s}_i (in particular rotations were used). This can be done since linear warping and isometries are both symmetries of the camera calibration problem. An extensive treatment of all the symmetries of the problem is given in [6] and [16].

Our first result consists in showing, in figure 2, the coordinates \tilde{s}_i for the pin-hole and fish-eye cameras for a retina of size 50. Figures 2(a) and 2(b) illustrate the calibration obtained with the displacement-based method in the calibration condition where the camera displacements were subject to 10% error. The results with zero error were

TABLE I

MEAN SCALED RELATIVE ERROR E_{sr} EVALUATED FOR THE DISPLACEMENT-BASED CALIBRATION METHOD. THE RESULTS ARE EXPRESSED IN NUMBER OF PIXELS.

Grid Size	Pin-Hole	Fish-Eye	Pin-Hole (Err. 10%)	Fish-Eye (Err. 10%)
20	$1 \cdot 10^{-3} \pm 1 \cdot 10^{-3}$	0.1 ± 0.1	0.02 ± 0.02	0.2 ± 0.2
30	$6 \cdot 10^{-4} \pm 7 \cdot 10^{-4}$	0.1 ± 0.1	0.02 ± 0.02	0.2 ± 0.3
40	$4 \cdot 10^{-4} \pm 5 \cdot 10^{-4}$	0.2 ± 0.2	0.04 ± 0.04	0.6 ± 0.7
50	$5 \cdot 10^{-4} \pm 5 \cdot 10^{-4}$	0.2 ± 0.2	0.04 ± 0.04	0.7 ± 0.8

better and are not illustrated. Figures 2(c) and 2(d) shows the results of the calibration obtained with the correlation-based algorithm. The main reason why our method outperforms Kuipers' is that our method exploits the fact that, when an agent performs a rigid movement, if two sensor happen to be in the same relative position with respect to the environment before and after the movement they will sense the same input. This is an actual property of space in the sense that it is independent of both the agent's sensors and the particular environment. In Kuipers's algorithm this is not the case: evaluating distances by comparing the values sensed by different sensors makes the distance evaluation dependent on both the environment and the sensors. Again, in this case distance estimation is accurate as long as the distances among photoreceptors are sufficiently small compared to the rapidity of change in the observed image. When the grid of photoreceptor is quite big and the image change quite rapidly, this condition is no longer satisfied, resulting in poor distance estimation. Empirical evidence for these assertions can be seen in figure 3, where we plot μ_{ij}^D and μ_{ij}^C against, the real distances among the relative directions of sight, $\rho(s_i, s_j)$. The figure presents the results on both the pin-hole (3(a) and 3(c)) and fish-eye (3(b) and 3(d)) cameras of size 40. It is clear from the figure that for bigger distances the value of μ^C is more or less constant, implying that a small error in the estimation of μ_{ij}^C leads to quite big error in the estimation of $\rho(s_i, s_j)$. This is not the case for the displacement-based algorithm, where the values of μ^D increase monotonically as a function of distance.

In order to evaluate the quality of the embedding, that is how accurate the estimation of distances among directions of sight was, we calculated the *mean scaled relative error* (MSRE from now on) E_{sr} , which is defined as follows:

$$E_{sr} = \min_{\alpha > 0} \frac{1}{n^2} \sum_{i,j=1}^n |\rho(s_i, s_j) - \alpha \rho(\tilde{s}_i, \tilde{s}_j)| \quad (6)$$

In this equation \tilde{s}_i are the estimated values of the directions of sight obtained with our algorithm. The distances among these are compared with the distances among the true values of the directions of sight s_i . The minimization over the multiplicative factor α is due to the fact that, as previously explained, linear warping is a symmetry of the camera calibration problem. The values of the MSRE for our calibration methods, expressed in number of pixels, are reported the table I. MSRE was calculated for the different sizes of the photoreceptor grid for the two kinds of cameras and in the two calibration conditions, both for the condition

with no error in camera displacement and when an error of the 10%. It is worth noting that the displacement-based algorithm performed well with a values of E_{sr} no greater then 0.7 pixels. This is quite meaning given that the maximum displacement considered for a photoreceptor grid of 50 was about 70 pixels (the size of the diagonal of the grid) with an error of about 7 pixels on the displacement. Finally, in figure 4 we present some pictures taken with the fish-eye camera before and after the calibration. These figures were obtained by calibrating the fish-eye camera with a photoreceptor grid of size 70. The results in the figure are from the calibration condition with no error added to the camera displacements. The original image shown to the camera is displayed in the upper left corner in all three figures (4(a), 4(b) and 4(c)). The image in the photoreceptor plane obtained with a non-calibrated fish-eye camera is shown in the upper right. The lower part of the three figures shows the images as taken with fish-eye camera calibrated with the displacement-based method (bottom left) or the correlation-based method (bottom right).

IV. CONCLUSIONS

Here we have presented a calibration technique with some very interesting characteristics that make it appealing for robotics. It is a completely unsupervised calibration technique which does not require a model of the camera so it can work for a big variety of optical devices. The calibration can be performed everywhere, with no requirement of particular calibration patterns or a particular environment. The smoothness of the images used in this work was similar to those of pictures taken with an off-the-shelf camera. Note, of course, that the smoothness/sharpness of images should not represent a constraint for our algorithm. In fact, since distances are evaluated by coupling photoreceptor which sense the same input before and after a displacement, our algorithm should work even when the calibration images are changing very quickly; possibly even random noise pictures could be used. Another important advantage of our algorithm is that it could potentially work with a camera with slightly different gain functions for each photoreceptor. This is because, in our algorithm the φ maps are built by associating photoreceptors with the best-correlated inputs. It is reasonable to believe that if the correlation between the inputs of two photoreceptors is the greatest, it will remain the same even if we systematically modify the inputs of the two photoreceptors using slightly different gain functions. This prediction will be precisely investigated in future work.

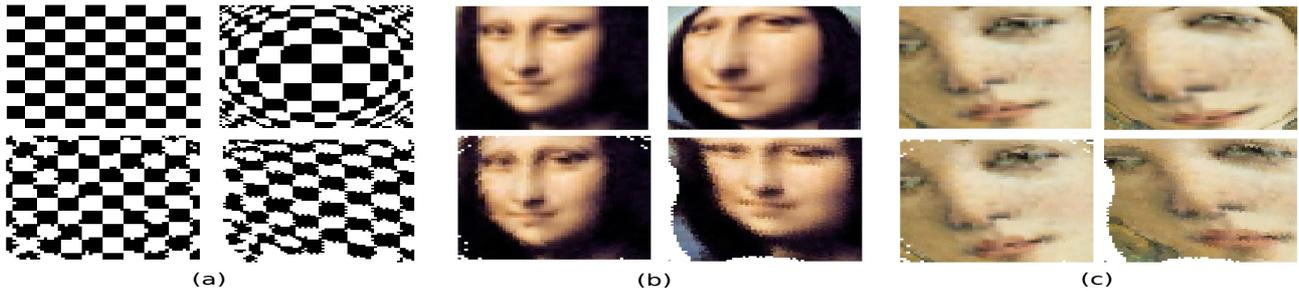


Fig. 4. In the upper left corner of (a), (b) and (c), the original images that the camera was pointing at. In the upper right, the images in the plane of photoreceptors for a non-calibrated fish-eye camera. In the lower left corner, the result of calibration with the displacement-based method, and in the lower right with the correlation-based method. Both pictures are drawn from the calibration of a fish-eye camera with a grid of photoreceptors of size 70. The results relative to the displacement-based algorithm are for the calibration condition where the displacements were not corrupted by any error. The camera, as stated above, was calibrated using grayscale images; here the results of the calibration are shown with coloured images to better illustrate the differences in the results.

The calibration technique we propose was tested in the condition of a camera performing particular kinds of movements, namely rigid displacements. These comport difficulties in comparing distances along the two directions as well as the angles between them and could be overcome by evaluating the φ functions also for camera rotations. Using just rigid displacements can hold only as an approximation for many real robot settings, but this approximation is not as brutal as it may seem. The movements of cameras used in robot applications are frequently rotations rather than translations as in the case of pan-tilt cameras. But it is definitely the case that the former can be good approximations of the latter for small angle rotations, which after all are the only kind required for our calibration algorithm. Moreover, the fact that our algorithm performs well even when the camera displacement is corrupted with error is promising for the success of the algorithm in a real setting.

The actual main limitation of our algorithm surely lies in its computational cost (the biggest retina calibrated in this paper has about 5000 photoreceptors while nowadays camera have millions of photoreceptors). In fact, establishing a precise calibration requires the evaluation of a quite large number of φ functions. Furthermore, evaluating each of the φ functions requires repeating the same movement a certain number of times. An additional study must be performed to understand under what condition and to what extent the number of movements performed affects the quality of the φ functions. In any case, the solution to the computational cost problem of our algorithm could even reside in the application of a hybrid algorithm, with the combined application of our technique and Kuipers' method. In such a mixed method our technique could be applied, for example, to only a subset of photoreceptors, perhaps a set which forms a regular grid with large step size. The evaluation of distances among nearby photoreceptors could be done using Kuipers' method.

ACKNOWLEDGMENT

In the present political and social context Montone and O'Regan wish to declare that, although very respectful of the ideas of all religious peoples, they grew up healthy and happy

without ever feeling the need to believe in the existence of any God.

REFERENCES

- [1] J.-Y. Bouguet, "Camera calibration toolbox for matlab," 2004.
- [2] Z. Zhang, "A flexible new technique for camera calibration," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [3] O. D. Faugeras, Q.-T. Luong, and S. J. Maybank, "Camera self-calibration: Theory and experiments," in *Computer Vision ECCV'92*. Springer, 1992, pp. 321–334.
- [4] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 8, pp. 1335–1340, 2006.
- [5] J. Weng, P. Cohen, and M. Herniou, "Camera calibration with distortion models and accuracy evaluation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 14, no. 10, pp. 965–980, 1992.
- [6] A. Censi and D. Scaramuzza, "Calibration by correlation using metric embedding from nonmetric similarities," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 10, pp. 2357–2370, 2013.
- [7] E. Grossmann, J. A. Gaspar, and F. Orabona, "Discrete camera calibration from pixel streams," *Computer Vision and Image Understanding*, vol. 114, no. 2, pp. 198–209, 2010.
- [8] J. Modayil, "Discovering sensor space: Constructing spatial embeddings that explain sensor correlations," in *Development and Learning (ICDL), 2010 IEEE 9th International Conference on*. IEEE, 2010, pp. 120–125.
- [9] J. Stober, L. Fishgold, and B. Kuipers, "Sensor map discovery for developing robots," in *AAAI Fall Symposium: Manifold Learning and Its Applications*, 2009.
- [10] D. Philipona, J. K. O'Regan, and J.-P. Nadal, "Is there something out there? inferring space from sensorimotor dependencies," *Neural computation*, vol. 15, no. 9, pp. 2029–2049, 2003.
- [11] D. Philipona, J. K. O'Regan, J.-P. Nadal, and O. Coenen, "Perception of the structure of the physical world using unknown multimodal sensors and effectors," in *Advances in neural information processing systems*, 2003, p. None.
- [12] A. V. Terekhov and J. K. O'Regan, "Space as an invention of biological organisms," *arXiv preprint arXiv:1308.2124*, 2013.
- [13] A. V. Terekhov and J. K. O'Regan, "Learning abstract perceptual notions: The example of space," in *Development and Learning and Epigenetic Robotics (ICDL-Epirob), 2014 Joint IEEE International Conferences on*, Oct 2014, pp. 368–373.
- [14] T. F. Cox and M. A. Cox, *Multidimensional scaling*. CRC Press, 2010.
- [15] D. Pierce and B. J. Kuipers, "Map learning with uninterpreted sensors and effectors," *Artificial Intelligence*, vol. 92, no. 1, pp. 169–227, 1997.
- [16] S. Agarwal, J. Wills, L. Cayton, G. Lanckriet, D. J. Kriegman, and S. Belongie, "Generalized non-metric multidimensional scaling," in *International Conference on Artificial Intelligence and Statistics*, 2007, pp. 11–18.