
DCCAL - Discrete Cameras Calibration using Properties of Natural Scenes

Milestone 3 Calibration Methodology

Funded by:

FCT Fundação para a Ciência e a Tecnologia
MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E ENSINO SUPERIOR

FCT PTDC/EEA-CRO/105413/2008
January 2010 - December 2012

July 2011
May 2012 (update)
January 2013 (update)

Topological Auto-Calibration of Central Imaging Sensors

R. Galego, R. Ferreira, A. Bernardino, E. Grossmann, J. Gaspar
Institute for Systems and Robotics / IST / UTL

Abstract

This paper departs from traditional calibration in the sense that the pixels forming the camera have an unknown topology. Previous works have shown that the statistical properties of natural scenes, and a uniform motion of a camera in 6 DoF, allow determining the topology of the camera. Here we show that there is a quasi-linear relationship between time-correlation and angular-distance inter-pixel, considering small angles and a simple scenario encompassing one bright light on a dark background as e.g. a full moon. The topology reconstruction algorithm is therefore constructed based on correlating time series (pixel streams) acquired by the pixels of the camera. Correlations are converted to inter-pixel distances using a fixed linear transformation. Distances are finally embedded on a plane using a manifold learning methodology. In particular we propose using the Landmark-Isomap learning methodology in order to deal with a large number of pixels (order of 10^4). Experiments both on simulated and real datasets have been conducted and shown promising results, namely that the theoretical derivations are accurate for a simple moonlight scenario, and tolerate significant deviations of the scenario such as a binarized natural scene.

1 Introduction

Traditional imaging sensors are formed by pixels precisely placed in a rectangular grid, and thus look like calibrated sensors for many practical purposes such as localizing local extrema, edges or corners. In contrast, the most common imaging sensors found in nature are the compound eyes, collections of individual photo cells which clearly do not form rectangular grids, but are very effective for solving various tasks at hand and thus have inspired the design of many artificial systems. Völkel *et al.* studied several types of eyes and discussed the miniaturization of imaging systems [15]. Neumann *et al.* [7] proposed a compound eye vision sensor for 3D ego motion computation. Recently, Micro-Electro-Mechanical Systems fabrication technologies were applied to build artificial compound eyes on planar surfaces [4].

In other words, novel fabrication technologies allow creating sensors with pixel arrangements (topologies) tuned for the tasks at hand. In the cases where the sensor topology is not a rectangular grid using traditional calibration methodologies [1, 16, 12] will not be possible. Hence, the question arising here is: how to calibrate sensors with unknown topologies? In the case that the sensors are mounted on mobile robots the question can be restated as: can we calibrate an unknown topology of a moving sensor just with the data acquired by the sensor?

Pierce and Kuipers have shown that it is possible to reconstruct the topology of a group of sensors just by knowing their output [10]. They use natural properties of an agent's world in order to infer the structure of its sensors. Olsson *et al.* improved the methodologies introduced by Pierce and Kuipers by adding information distances and, in particular, Hamming metrics [8, 9]. They compute the position of several sensors of a Sony Aibo robot, which has, among other sensors, one camera sub-sampled to 8×8 pixels. Hyvarinen *et al.* shown that imaging natural scenes allow defining a *neuronal topography* using Independent Subspace Analysis [6]. Recently, Grossmann *et al.* [5] proposed a method for calibrating a central imaging sensor based on a number of photocells. They need to know (estimate) a priori a function curve relating correlation (or information-distance) and distance-angles. Their algorithm has been tested on a small set of pixels (photocells), about one hundred, as otherwise the computation time and memory would be too large.

In this work we want to do auto-calibration of central sensors with a number of pixels orders of magnitude larger than [9, 5]. We approach the computational complexity with Multi Dimensional Scaling (MDS) like algorithms. A relatively old but very effective in the presence of noise free data is the Classical MDS [2], based on Euclidean distances. Its goal is to find a representation of a data set on a given dimensionality from the knowledge of all interpoint distances. Several new algorithms evolved from MDS, such as Isomap [14], where geodesic distances induced by a neighborhood graph are used instead of Euclidean distances. More recently Landmark Isomap [11] was introduced, which uses only a subset

of the all-to-all distances used in Isomap and has been proven to work on large scale datasets (millions of data points) [13] and constitute therefore a promising research direction.

The structure of the paper is the following: in Sec.2 we study a simple black and white scenario and show there is a linear relationship between the correlation of the time series acquired by pairs of pixels and the inter-pixel angle; in Sec.3 we describe Landmark-Isomap applied to topological calibration and propose a methodology for choosing a coordinate frame for the imaging sensor; in Sec.4 we show some experimental results, and finally in Sec.5 we draw some conclusions.

2 Correlation of Pixel Streams

2.1 Correlation vs Angular Distances

Let $f \in \mathbb{R}^k$ and $g \in \mathbb{R}^k$ be the aligned pixel streams of two pixels located on a sphere at position $\mathbf{p} \in \mathbb{S}^2$ and $\mathbf{q} \in \mathbb{S}^2$. Considering the normalized cross correlation

$$C(f, g) = \frac{\mathbb{E}[fg] - \mathbb{E}[f]\mathbb{E}[g]}{\sqrt{(\mathbb{E}[f^2] - \mathbb{E}^2[f])(\mathbb{E}[g^2] - \mathbb{E}^2[g])}} \quad (1)$$

Grossmann *et al.*[5] showed experimentally that there is a relationship between the normalized cross correlation and the angular distance between pixel streams, $d(\mathbf{p}, \mathbf{q}) = \text{acos}(\mathbf{p}^T \mathbf{q})$.

Property 1. *Correlation invariant to the shuffling of time series*

Here we propose to better understand the relationship between correlation and angular distance in a particular simple case where: (i) the camera is in the center of a unit sphere ($R = 1$) which is black everywhere except in a hub-cap corresponding to the projection of a white circle with radius r tangent to the sphere, (ii) the integration area of each pixel of the camera collapses to a single point, and (iii) the camera motion covers uniformly the rotation configuration space $\mathbb{SO}(3)$ and acquires an infinite time series (pixel stream) with each of the pixels.

Equation (1) shows that to compute the theoretical correlation value, one needs only to compute $\mathbb{E}[f]$, $\mathbb{E}[g]$, $\mathbb{E}[f^2]$, $\mathbb{E}[g^2]$ and $\mathbb{E}[fg]$. Since the camera motion is uniform and the time sequences are infinite, there is no difference in the expected values of the isolated pixel stream, i.e. $\mathbb{E}[f] = \mathbb{E}[g]$ and $\mathbb{E}[f^2] = \mathbb{E}[g^2]$. Without loss of generality it is assumed that each pixel reports a value of 1 when it is pointed towards the white circle and a value of -1 when pointed towards a black area. Note that the definition of correlation is invariant to the choice of these values. This particular choice leads to $\mathbb{E}[f^2] = \mathbb{E}[g^2] = 1$. We are left with the task of computing $\mathbb{E}[f]$ and $\mathbb{E}[fg]$.

Expected value of a pixel stream, $\mathbb{E}[f]$ - Since the camera pose distribution is uniform, the probability of a pixel observing the white circle corresponds to the area of this circular section ¹ $A_d = 2\pi R(R - \sqrt{R^2 - r^2})$ divided by the total surface area of the sphere ($A_s = 4\pi R^2$):

$$p = \mathbb{P}[\text{Pixel observing white}] = \left(2\pi(1 - \sqrt{1 - r^2})\right) / (4\pi) = \left(1 - \sqrt{1 - r^2}\right) / 2 \quad (2)$$

and thus

$$\mathbb{E}[f] = 1 \cdot p + (-1)(1 - p) = -\sqrt{1 - r^2}. \quad (3)$$

In the case of small values of r , Eq.2 can be approximated by $p \approx r^2/4$ which corresponds to using the area of the circle, $A_c = \pi r^2$, instead of the area of the circular cap A_d . If the approximate formula is used, the expected value is $\mathbb{E}[f] \approx r^2/2 - 1$.

Cross correlation, $\mathbb{E}[fg]$, approximated value - To compute $\mathbb{E}[fg]$ first notice that the pointwise result of fg is either 1, when both pixels observe the interior of the circle or none of the pixels do, or -1 when one pixel observes the interior of the circle and the other does not.

Note: relate $\mathbb{E}[fg]$ with convolution on a sphere surface

Here we approximate the true value of $\mathbb{E}[fg]$ by assuming that the observed white circle is small, thus we can think of the necessary areas on the tangent planes v.s. on the surface of the sphere. Figure 1 shows two pixels \mathbf{p} and \mathbf{q} separated by a distance d on this tangent plane. We know that a pixel observes white whenever the center of the projected white circle falls within a distance r of the pixel (within circle

¹ http://en.wikipedia.org/wiki/Spherical_cap

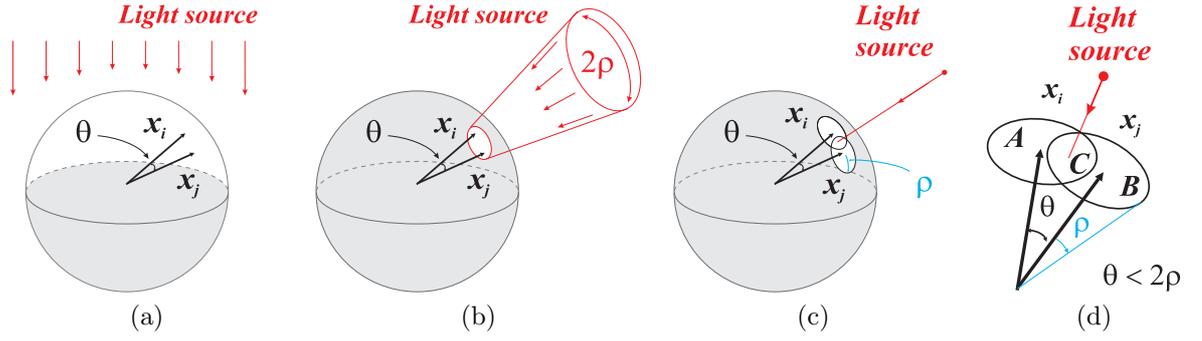


Fig. 1: Illumination and the information acquired by pixels x_i and x_j . A light source like the sun is seen as an illuminating hemisphere around the camera (a). A light source like a full-moon is seen as an illuminating hubcap (b). Instead of infinitesimal pixels, one can consider enlarged pixels and an infinitesimal illumination (c). The intersection of the pixel areas $C = A \cap B \neq \emptyset$ if $\theta < 2\rho$ (d).

A in the figure or within circle B, depending on the considered pixel). Thus the probability of exactly one pixel observing the white circle is

$$\frac{\text{Area}(A) + \text{Area}(B) - 2 \text{Area}(C)}{\text{Area}(\text{Sphere})} \quad (4)$$

where the intersection area C is removed since the center of the circle falling within this region means both pixels would measure white. Again note that here the numerator is approximated. Since $\text{Area}(A) = \text{Area}(B) = 2\pi r^2$ and, if the two circles intersect ², $\text{Area}(C) = 2r^2 \arccos(\frac{d}{2r}) - \frac{d}{2} \sqrt{4r^2 - d^2}$ one has that in the case of $2r > d$ the probability is

$$p = \mathbb{P}[\text{Pixel observing white}] \approx \left(2\pi r^2 - 4r^2 \arccos\left(\frac{d}{2r}\right) + d\sqrt{4r^2 - d^2} \right) / (4\pi), \quad (5)$$

or, in the case that the two circles do not intersect,

$$p = \mathbb{P}[\text{Pixel observing white}] \approx (2\pi r^2) / (4\pi) = r^2 / 2. \quad (6)$$

Finally, one obtains the approximated value of the cross correlation as

$$\mathbb{E}[fg] = (-1)p + (1)(1 - p) = 1 - 2p \quad (7)$$

$$\approx \begin{cases} 1 - \frac{2\pi r^2 - 4r^2 \arccos(\frac{d}{2r}) + d\sqrt{4r^2 - d^2}}{2\pi} & \text{if } 2r > d \\ 1 - r^2 & \text{if } 2r \leq d. \end{cases} \quad (8)$$

Cross correlation, $\mathbb{E}[fg]$, true value - TBA - Area of A and B is already computed, its Area of C that is hard(er).

The area of intersection of two equal sized circular sections (radius r) on the sphere, at an angular distance d , can be computed as

$$A_i = \iint_C d\omega \quad (9)$$

where ω is the area form on the surface of the sphere. To compute this integral we'll choose as coordinates the projection from the center of the sphere of the sphere's surface on a tangent plane touching the sphere at the center of one of the circular sections.

The coordinate transformation from the tangent plane to the sphere is thus

$$f(x, y) = \begin{bmatrix} \frac{x}{\sqrt{x^2 + y^2 + 1}} \\ \frac{y}{\sqrt{x^2 + y^2 + 1}} \\ \frac{1}{\sqrt{x^2 + y^2 + 1}} \end{bmatrix} \quad (10)$$

² <http://mathworld.wolfram.com/Circle-CircleIntersection.html>

from the jacobian determinant of this coordinate transformation, the integral 9 can be computed as

$$A_i = \iint_{f^{-1}(C)} (x^2 + y^2 + 1)^{-3/2} dx \wedge dy \quad (11)$$

$$A_i = 4 \int_{\tan(d/2)}^{\frac{r}{\sqrt{1-r^2}}} \int_{-\sqrt{\frac{r^2}{1-r^2}-y^2}}^0 (x^2 + y^2 + 1)^{-3/2} dx \wedge dy \quad (12)$$

The area is thus

$$A_i = 2\pi \left(1 - \sqrt{1-r^2}\right) \quad (13)$$

$$- 4 \operatorname{asin} \left(\frac{\tan(d/2)}{r\sqrt{\tan^2(d/2)+1}} \right) \quad (14)$$

$$- 4\sqrt{1-r^2} \operatorname{asin} \left(-\sqrt{\frac{1-r^2}{r^2}} \tan(d/2) \right) \quad (15)$$

$$= 2\pi (1 - \cos(\rho)) \quad (16)$$

$$- 4 \operatorname{asin} \left(\frac{\sin(d/2)}{\sin(\rho)} \right) \quad (17)$$

$$- 4 \cos(\rho) \operatorname{asin} \left(-\frac{\tan(d/2)}{\tan(\rho)} \right) \quad (18)$$

$$(19)$$

where $r = \sin(\rho)$ (angular angle).

This can now be inserted in equation 4 as the area of C , while the area of A and B are given by A_d . Thus,

$$p = \mathbb{P}[\text{Pixel observing white}] \quad (20)$$

$$= \frac{8 \operatorname{asin} \left(\frac{\sin(d/2)}{\sin(\rho)} \right) + 8 \cos(\rho) \operatorname{asin} \left(-\frac{\tan(d/2)}{\tan(\rho)} \right)}{4\pi} \quad (21)$$

$$= \frac{2}{\pi} \operatorname{asin} \left(\frac{\sin(d/2)}{\sin(\rho)} \right) + \frac{2}{\pi} \cos(\rho) \operatorname{asin} \left(-\frac{\tan(d/2)}{\tan(\rho)} \right) \quad (22)$$

$$(23)$$

thus

$$\mathbb{E}[fg] = 1 - \frac{4}{\pi} \operatorname{asin} \left(\frac{\sin(d/2)}{\sin(\rho)} \right) - \frac{4}{\pi} \cos(\rho) \operatorname{asin} \left(-\frac{\tan(d/2)}{\tan(\rho)} \right) \quad (24)$$

2.1.1 Computing the correlation

Now that all the values are available, $C(f, g)$ can be computed as in expression (1).

Approximated -

$$C(f, g) \approx \begin{cases} \frac{1 - \frac{2\pi r^2 - 4r^2 \operatorname{acos}(\frac{d}{2r}) + d\sqrt{4r^2 - d^2}}{2\pi} - (\frac{r^2}{2} - 1)^2}{1 - (\frac{r^2}{2} - 1)^2} & \text{if } 2r > d \\ \frac{1 - r^2 - (\frac{r^2}{2} - 1)^2}{1 - (\frac{r^2}{2} - 1)^2} & \text{if } 2r \leq d \end{cases} \quad (25)$$

True -

$$C(f, g) = \frac{\mathbb{E}[fg] - (1 - r^2)}{\sqrt{(1 - (1 - r^2))(1 - (1 - r^2))}} = \frac{\mathbb{E}[fg] - (1 - r^2)}{r^2} \quad (26)$$

$$C(f, g) = \begin{cases} \frac{-\frac{4}{\pi} \operatorname{asin} \left(\frac{\sin(d/2)}{\sin(\rho)} \right) - \frac{4}{\pi} \cos(\rho) \operatorname{asin} \left(-\frac{\tan(d/2)}{\tan(\rho)} \right) + \sin^2(\rho)}{\sin^2(\rho)} & \text{if } 2\rho > d \\ 1 - \frac{2(\cos(\rho) - 1)}{\sin^2(\rho)} & \text{if } 2\rho \leq d \end{cases} \quad (27)$$

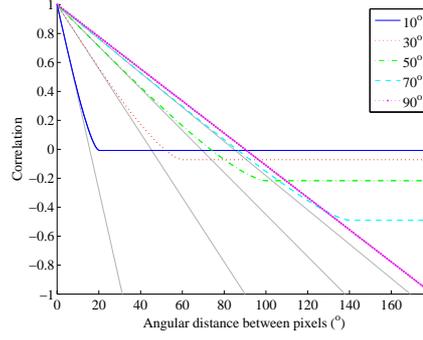


Fig. 2: Correlation v.s. angular distance between pixels for several observed circle sizes. The tangents to each curve at the origin are overlayed in light gray.

Taylor expansion The 4th order Taylor expansion of the correlation function near the origin is

$$C(f, g) \approx 1 - \frac{2}{\pi \sin(\rho)} d - \left(\frac{1}{12\pi \sin(\rho)} - \frac{1}{12\pi \sin^3(\rho)} \right) d^3 \quad (28)$$

Property 2. Correlation, $C(f, g)$, is a linear function of the inter pixel distance, $d(p, q)$, when $\rho = \pi/2$.

Demo:

Observation 1. Correlation, $C(f, g)$, is approximately a linear function of the inter pixel distance, $d(p, q)$, when $d \approx 0$.

As shown in figure 2, and in the Taylor expansion.

3 Auto-Calibration Methodology

For the particularly simple scenery described in section ??, it is clear that for correlation values above 0.5 the look up table needed in [5] can be replaced by a single slope parameter dependent on the size of the observed hubcap. We propose to further drop this requirement for small sensors by embedding the locations in a plane up to a scale factor. The proposed method consists of directly converting the obtained correlation values to distance values and then applying the landmark Isomap algorithm [11] to obtain the reconstruction. Hence we will consider that the distance between two pixels whose correlation is above 0.5 is given by $d(\mathbf{x}_i, \mathbf{x}_j) = 1 - C(f_i, f_j)$. In the following we provide a short description of the Landmark Isomap algorithm.

MDS and Landmark Isomap - The classical Multi-Dimensional Scaling (MDS) [2] is based on euclidean distances and its goal is to find the optimal representation of a data set, X , such that:

$$d^2(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i - \mathbf{x}_j, \mathbf{x}_i - \mathbf{x}_j \rangle = \langle \mathbf{x}_i, \mathbf{x}_i \rangle - 2\langle \mathbf{x}_i, \mathbf{x}_j \rangle + \langle \mathbf{x}_j, \mathbf{x}_j \rangle \quad (29)$$

Collecting all squared distances in a matrix $D^2 = [d^2(\mathbf{x}_i, \mathbf{x}_j)]$ this can be transformed to a matrix of inner products by inverting the previous linear relation and forcing the resulting embedding to have zero mean [3]. More precisely, a matrix of inner products G is obtained from D^2 through the transformation $G = -JD^2J/2$, where $J = (I - 1/n)$, I is the $n \times n$ identity matrix and n is the number of the elements of the data. Next, one observes that if the desired point embedding is collected in a matrix $X = [\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_n]$, then

$$G = \begin{bmatrix} \langle \mathbf{x}_1, \mathbf{x}_1 \rangle & \dots & \langle \mathbf{x}_1, \mathbf{x}_n \rangle \\ \vdots & \ddots & \vdots \\ \langle \mathbf{x}_n, \mathbf{x}_1 \rangle & \dots & \langle \mathbf{x}_n, \mathbf{x}_n \rangle \end{bmatrix} = X^T X. \quad (30)$$

X can thus be obtained (reconstructed) up to a unitary transformation using an SVD decomposition, as $G = X^T X = U\Sigma U^T = U\sqrt{\Sigma}\sqrt{\Sigma}U^T$, and thus $X^T = U\sqrt{\Sigma}$.

Property 3. If distances are multiplied by a scale factor, $d_1 = \alpha d$, then the resulting topology is scaled by the same factor, $X_1 = \alpha X$.

Proof. Suppose all distances are affected by a gain α , $d_1 = \alpha d$, then the matrix of squared distances will be $D_1^2 = \alpha^2 D^2$. The inner product matrix will have a gain of α^2 since $G_1 = \alpha^2 J D^2 J / 2$ which will make the SVD of matrix $G_1 = U(\alpha^2 \Sigma)U^T$. The solution of the algorithm will then be $X_1^T = U\sqrt{\alpha^2 \Sigma}$. Since $X^T = U\sqrt{\Sigma}$ one concludes that $X_1 = X\alpha$ which proves that a gain in the distances changes the topology of the sensor only by a scale factor as required. \square

The classical MDS works well when the distances are Euclidean and when the structures are linear, however, when the manifolds are nonlinear, the classical MDS fails to detect the true dimensionality of the data set. Isomap is built on classical MDS but instead of using Euclidean distances it uses an approximation of geodesic distances [14]. These geodesic distance approximations are defined as a series of hops between neighboring points in the Euclidean space using a shortest path graph algorithm such as Dijkstra's.

Isomap has two computational bottlenecks, namely the memory and time complexity of computing an all-to-all distance matrix, $n \times n$ for a n pixels camera, and computing its eigenvalues. Landmark Isomap improves both inefficiencies [11]. Instead of using all the data points Landmark Isomap proposes using just k randomly selected points (landmarks) with $k \ll n$. The embedding is done the same way as in MDS but using only the k landmarks. The distance matrix, D is now just a $k \times n$ matrix. After embedding the landmarks, $K = \sqrt{\Sigma}U^T$, where K collects the k reconstructed landmark locations, one can embed the remaining points:

$$X = \frac{1}{2}K^*(\overline{D_k} - D_k) \quad (31)$$

where $K^* = [u_1^T/\sqrt{\Sigma_1} \dots u_s^T/\sqrt{\Sigma_s}]^T$ is the pseudo-inverse transpose of K , s is the size of the dimensionality space, D_k is the distance matrix from the landmarks to the complete data, and $\overline{D_k}$ is the mean of the columns of D_k .

In our particular case, this algorithm is used to provide a pixel embedding given the inter-pixel distances estimated from the pixel stream correlations.

Choosing the Imaging Coordinate System - As referred, MDS (and derived methods) provide a reconstruction of the vectors collected in X up to a unitary transformation. Assuming that the camera is mounted on a mobile robot, we propose to fix the unitary transformation in accordance with the motion degrees of freedom of the robot.

Having reconstructed the topology of the imaging sensor allows doing 2D interpolation and therefore computing (approximated) directional derivatives and finding feature points using standard image processing techniques. Then, considering for example that a camera has experienced from t_1 to t_2 a leftwards pan motion and from t_3 to t_4 an upwards tilt motion, where t_i denote timestamps, allows computing two median optical-flows (or disparities), v_1 and v_2 . The two flow vectors allow therefore setting the coordinates of a pixel location to be first horizontal, growing right, and the second to be vertical, growing down:

$$X_f = TX = [\hat{v}_1 \ \hat{v}_2]^{-1}X \quad (32)$$

where \hat{v}_1 and \hat{v}_2 denote normalization to unit length of v_1 and v_2 . Note that noise prevents perfect orthogonality, i.e. $v_1^T v_2 \neq 0$, in which case we rotate both vectors in opposing directions to meet orthogonality. Having $v_1^T v_2 = 0$ with nonzero v_1 and v_2 , implies $|\det(T)| = 1$, where $\det(T) = -1$ indicates a mirroring effect found in the reconstructed topology.

Calibration Methodology - Summarizing the previous sections, estimating and embedding the topology of a central imaging sensor involves acquiring a set of images observing a bright circle faraway, e.g. the full moon. Since the correlation of pixel-streams, $C(f_i, f_j)$ is invariant to shuffling of the time-series (as long as both time series are affected by the same shuffling), these images can be acquired either as a continuous sequence (i.e. a video) or as discrete individual images. In the end one wants to obtain the embedded pixel locations, $X_f = [\mathbf{x}_1 \ \mathbf{x}_2 \ \dots \ \mathbf{x}_N]$. The required steps are the following: (i) Data binarization using a fixed threshold such that each pixel stream value is either 1 or -1 . (ii) Computing the normalized correlation between all the pixel-streams using Eq.1. (iii) Converting the inter-pixel correlations into distances, using the linear transformation $d(\mathbf{x}_i, \mathbf{x}_j) = 1 - C(f_i, f_j)$ (based on properties ?? and 3). (iv) Using Landmark Isomap to compute the topology of the sensor. (v) (optional in case an external reference frame is available) Choosing a coordinate system for the camera based on the supporting robot motion (Eq.32).

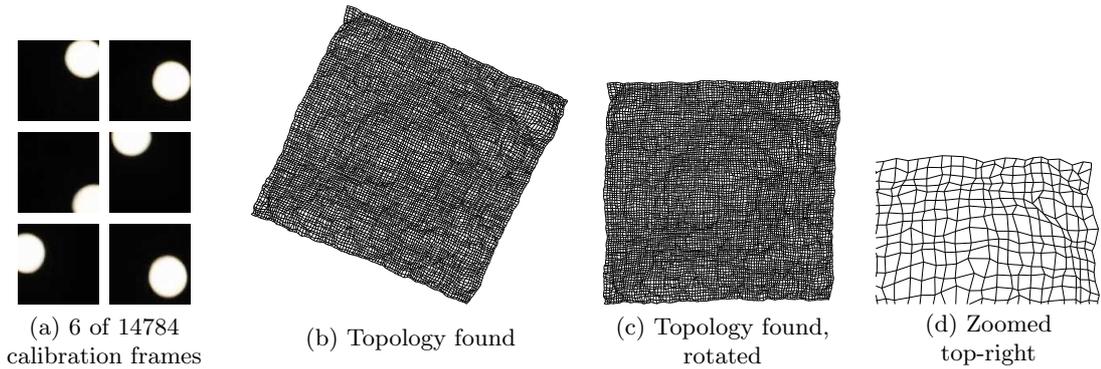


Fig. 3: Moon images used to estimate the topology of a 100×100 sensor (a). Estimated topology after random permutation of the pixel-streams (b,c,d).

4 Results

In order to test the proposed topology estimation methodology two experiments have been conducted using a Nikon D5000 camera in video mode, selecting just a central region of 100×100 pixels, and thus having the ground truth of a sensor composed by square pixels forming a regular square grid.

Topology estimation In the first experiment the camera was pointed to the moon, in a full-moon night, to obtain calibration data (see Fig. 3(a)). The data acquisition was performed at 24fps for about ten minutes (14784 frames), while panning and tilting the camera. Figures 3(b,c) show the estimated topology, approximately forming a regular square grid, close to the ground truth.

Robustness to pixel shuffling and illustration of an horizontal mirror effect and its correction A test image, acquired in daylight (Fig. 4(d)), was then used to illustrate more clearly that the sequencing of the pixel-streams (see pixel permutations in Figs. 4(e,f,g,h)) does not influence the perceptual quality of the estimated topology and image reconstruction (Figs. 4(i,j,k)). Despite having obtained the results in Figs. 4(i,j,k) with different calibrations, and thus subject to different random selections of landmark pixels, the differences of the estimated topologies are small. Using a 2D Procrustes, to register the three reconstructed topologies with a square 1-pixel-steps grid, resulted in inter-pixel (four nearest neighbors) distance-error distribution with a standard deviation of 0.566, 0.563 and 0.563 pixels, respectively.

Topology estimation using the garden sequence In the second experiment, the main purpose is to explore more general calibration scenarios, while keeping the topology estimation accurate. In the moon sequence, edges are clearly defined and have directions distributed uniformly despite of the discretization due to the non-infinitesimal pixel size. Hence we considered the garden scenario, Fig. 5(a,b,c,d), where the vegetation also provides many edge directions. For this data set we filmed about twelve minutes, at 24fps, having acquired 17448 frames. In this case we do pan and tilt motions, as well as roll and translation. Figure 5(g) shows the topology reconstruction considering binary level pixel-streams. Some images used by the algorithm can be seen in Figs. 5(e,f). Despite the complexity of scene texture we do not use a look up table as required in [5]. Note that Fig. 5(g) is the direct result of the Landmark Isomap, while Fig. 5(h) is the result after using Eq.32, and thus show a detected and corrected mirror effect.

5 Conclusions and Future Work

In this paper we consider cameras as simple collections of pixels, arranged with an unknown topology, that one wants to estimate. By assuming that the cameras are mobile, one can obtain the topology from the correlation of the pixel-streams. In particular, we demonstrated that in a simple scenario where the camera is in the center of a spherical-surface, half-white and half-black, the correlation of pixel-streams is linearly related with the inter-pixel angle. In the cases where the white hub-cap is smaller than a hemisphere, we show that the relationship between correlation and inter-pixel angle is approximately linear for high correlation values. Combining the inter-pixel angle estimation, directly from correlation, with manifold learning for embedding all the inter-pixel angles, has proved to be an effective methodology for topology estimation. In addition, we presented a methodology for detection/correction the mirror effect.

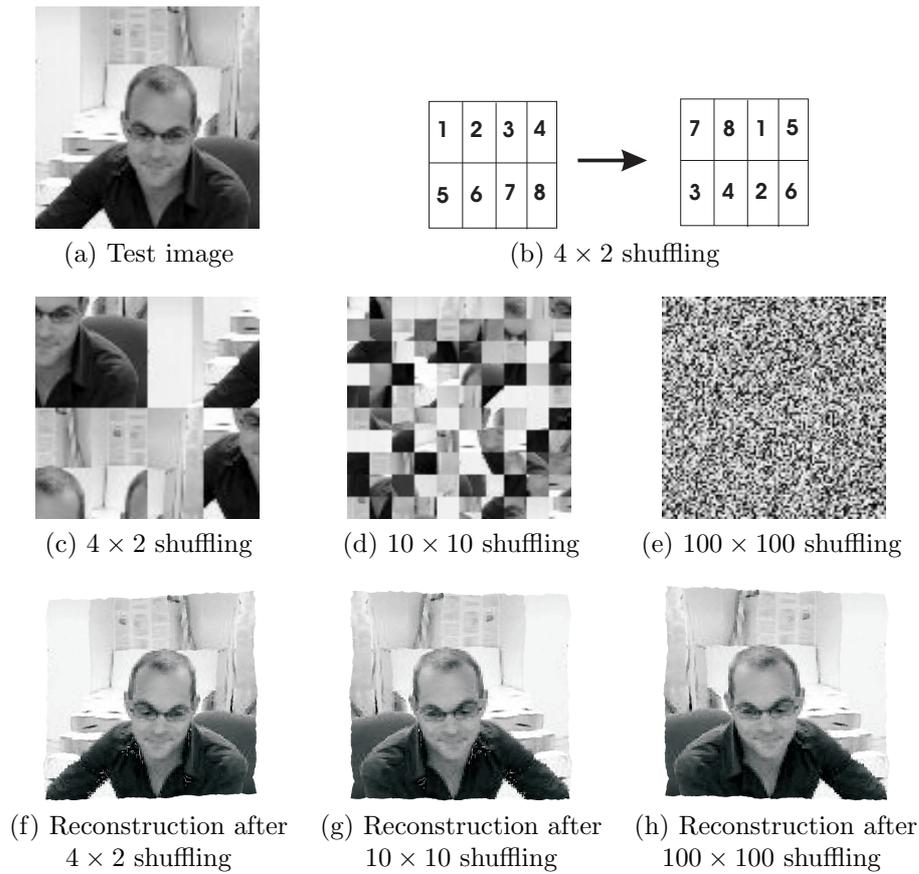


Fig. 4: Topology estimation applied to the reconstruction of a shuffled 100×100 sensor. Test image before shuffling (a). The pixels of the sensor are shuffled in (i) 4×2 blocks, (ii) 10×10 blocks, or (iii) 100×100 (all) pixels, as illustrated on the test image, (b,c), (d) or (e), respectively. Each of the three shufflings is then subject to topology estimation, and the resulting topology applied to reconstruct the test image (f,g,h).

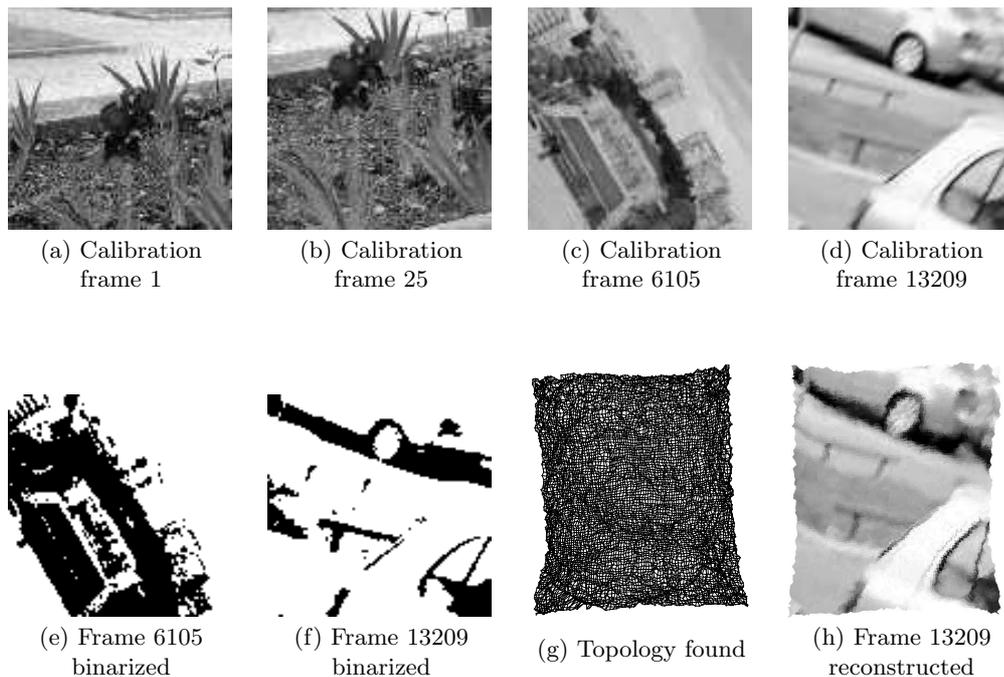


Fig. 5: Topology estimation applied to 17448 images random images of the University campus

effect inherent to the manifold learning modalities presented. Future work will focus on calibrating cameras combining optic fibers with a conventional CCD/CMOS sensor that implicitly impose the need of topological calibration.

References

- [1] L. Agapito, R. Hartley, and E. Hayman. Linear calibration of a rotating and zooming camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 15–21. IEEE Computer Society, Fort Collins, Colorado, USA, June 1999.
- [2] T.F. Cox and M.A.A. Cox. *Multidimensional scaling*. Chapman & Hall/CRC, 2001.
- [3] Jon Dattorro. *Convex Optimization and Euclidean Distance Geometry*. Meboo Publishing USA, 2010.
- [4] Si Di, Hui Lin, and Ruxu Du. An artificial compound eyes imaging system based on mems technology. In *Robotics and Biomimetics (ROBIO), 2009 IEEE International Conference on*, pages 13–18, dec. 2009.
- [5] Etienne Grossmann, Jose Gaspar, and Francesco Orabona. Discrete camera calibration from pixel streams. *Computer Vision and Image Understanding (Special issue on Omnidirectional Vision, Camera Networks and Non-conventional Cameras)*, 114(2):198–209, February 2010.
- [6] Aapo Hyvriinen, Jarmo Hurri, and Patrick O. Hoyer. *Natural Image Statistics: A Probabilistic Approach to Early Computational Vision*. Springer Publishing Company, Incorporated, 1st edition, 2009.
- [7] Jan Neumann, Cornelia Fermuller, Yiannis Aloimonos, and Vladimir Brajovic. Compound eye sensor for 3d ego motion estimation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '04)*. IEEE, April 2004.
- [8] L. Olsson, C. L. Nehaniv, and D. Polani. Sensory channel grouping and structure from uninterpreted sensor data. In *NASA/NoD Conference on Evolvable Hardware*, 2004.
- [9] Lars Olsson, Chrystopher L. Nehaniv, and Daniel Polani. Measuring informational distances between sensors and sensor integration. Technical report, In: *Artificial Life X*, 2006.
- [10] David Pierce and Benjamin J. Kuipers. Map learning with uninterpreted sensors and effectors. *Artif. Intell.*, 92(1-2):169–227, May 1997.
- [11] Vin De Silva and Joshua B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In *Advances in Neural Information Processing Systems 15*, pages 705–712. MIT Press, 2003.
- [12] S. N. Sinha and M. Pollefeys. Towards calibrating a pan-tilt-zoom camera network. In *Department of Computer Science, University of North Carolina at Chapel Hill*, pages 91–110, 2006.
- [13] A. Talwalkar, S. Kumar, and H. Rowley. Large-scale manifold learning. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, june 2008.
- [14] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290, 2000.
- [15] R. Völkel, M. Eisner, and K. J. Weible. Miniaturized imaging systems. *Microelectron. Eng.*, 67-68(1):461–472, June 2003.
- [16] Zhengyou Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *in ICCV*, pages 666–673, 1999.