Computer Vision and Image Understanding 125 (2014) 172–183

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

3D to 2D bijection for spherical objects under equidistant fisheye projection ☆

Aamir Ahmad *, João Xavier, José Santos-Victor, Pedro Lima

Institute for Systems and Robotics, Instituto Superior Técnico, Lisboa, Portugal

ARTICLE INFO

Article history:
Received 6 November 2013
Accepted 5 April 2014
Available online 16 April 2014

Keywords:
Equidistant projection
Fisheye lens
Spherical object detection
Omnidirectional vision
3D detection

ABSTRACT

The core problem addressed in this article is the 3D position detection of a spherical object of known-radius in a single image frame, obtained by a dioptric vision system consisting of only one fisheye lens camera that follows equidistant projection model. The central contribution is a bijection principle between a known-radius spherical object's 3D world position and its 2D projected image curve, that we prove, thus establishing that for every possible 3D world position of the spherical object, there exists a unique curve on the image plane if the object is projected through a fisheye lens that follows equidistant projection model. Additionally, we present a setup for the experimental verification of the principle's correctness. In previously published works we have applied this principle to detect and subsequently track a known-radius spherical object.

1. Introduction

1.1. Motivation

In recent years, omni-directional vision involving cata-dioptric [1] and dioptric [2] vision systems has become one of the most sought-after technology being employed in mobile robots. The primary reason of this success is a much wider field of view compared to a perspective projection lens-based camera. A dioptric vision system uses only a fisheye lens camera following equidistant projection model instead of a combination of a perspective projection camera and a parabolic mirror arrangement, often found in cata-dioptric vision systems. The calibration parameters of a cata-dioptric system are susceptible to variations, owing to physical vibrations and impact force when such a system is installed on a mobile robot platform and subjected to fast motion and unexpected collisions. The dioptric system overcomes this problem by having a more robust physical setup. Although we focus on a dioptric vision system (DVS) in this article, the theory developed here can be easily modified for cata-dioptric vision systems (CVS) by using the appropriate projection model and following the steps similar to the ones described further in this article.

1.2. Related work

Spherical object position detection is a functionality required by innumerable applications in the areas ranging from mobile robotics [1], biomedical imaging [3,4], machine vision [5] to space-related robotic applications [6]. In biomedical imaging, often the goal is to automatically detect spherical shaped anomalies that signify or precede a certain medical condition, e.g., detection of brain microbleeds in Alzheimer's disease [7]. Although in such applications the sphere detection usually assumes 3D image data (voxel grid, etc.) enabling direct application of Hough transform-like (HT) methods [3], the sheer amount of data that needs to be analyzed calls for faster methods of sphere detection. Authors in [4,8] explore the property of rotational invariance of spherical objects to develop highly precise and accurate sphere detection methods. However, those are applicable only on 3D image data as well as require considerable amount of training. On the other hand, some researchers have focused on improving the generalized HT-based methods in higher dimensional spaces. Authors in [5] introduce a concept of axial discretization to detect parts of a sphere in 3D image data using generalized HT but the proposed method would still be infeasible for real-time systems that require sphere detection at ~10–20 Hz. An HT-based detection has also been presented in [9] which utilizes Microsoft Kinect-generated 3D point clouds but it suffers similarly as [5] in computation time.

In mobile robotics, detection and tracking of spherical objects has gained substantial attention [10,11], including the use of omni-directional vision systems. In [1], the authors present a circular HT based spherical object detection using a CVS. Although they

☆ This paper has been recommended for acceptance by Andrea Prati.

* Corresponding author.

E-mail addresses: aahmad@isr.ist.utl.pt (A. Ahmad), jxavier@isr.ist.utl.pt (J. Xavier), jasv@isr.ist.utl.pt (J. Santos-Victor), pal@isr.ist.utl.pt (P. Lima).

http://dx.doi.org/10.1016/j.cviu.2014.04.004
1077-3142/© 2014 Elsevier Inc. All rights reserved.

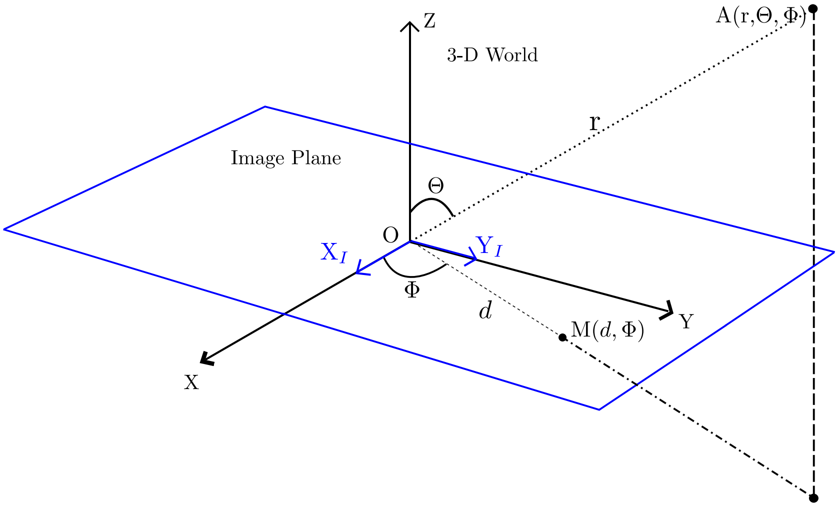


Fig. 1. 3D World frame to image frame transformation due to equidistant projection model. The camera lens is facing upwards in the positive Z direction in this figure.

use a single camera to perform the detection in real-time, their algorithm assumes that the object maneuvers on a fixed and known ground plane hence the detection is performed in a 2D world space. They extend the detection to the 3D space by using a stereo vision system (SVS) on their robots consisting of a CVS and a perspective camera installed on the same robot [12]. A similar approach is taken by authors in [11], where an SVS comprising of a CVS and a perspective camera looking in the forward direction is used. Here, instead of using an HT-based detection, the authors use a set of heuristics such as the color of the projected blobs in the image, and the roundness of the blobs, to detect pixel position in each individual image from both the cameras and later process them using the SVS calibration parameters to obtain the spherical object's 3D position. The authors in [13] present an interesting method of using the rotational invariance of the spherical object in a structure tensor technique to remove the use of color in the detection process. However, they still use a single perspective camera and perform detection only in the 2D world space, assuming that the spherical object is on a fixed and known ground plane. Furthermore, in all these works, the radius of the spherical object is known beforehand, which is an acceptable assumption for a variety of mobile robot tasks.

Most of the existing methods in literature make use of an SVS to detect the spherical object's position in 3D even when its radius is known beforehand. Our work however shows that if the radius is known, then it is possible to perform the detection in the 3D world space using only a single camera image. Some of the authors of this article have presented earlier such a method, using a color histogram mismatch-based algorithm [14]. However, the concept of uniquely detecting the 3D position using only a single camera image was not theoretically proven and the method was computationally heavy, often making it unsuitable for real-time applications.

1.3. Contribution

Since it is not obvious and to the best of our knowledge neither proposed nor proved in the existing literature that the 3D position of a known-radius sphere can be estimated using a single image from a fisheye lens-based camera following the equidistant projection model, we propose and prove a bijection principle for it, which is the novel contribution of this article. The bijection principle proposes that for every given position of a known-radius spherical object in the 3D world, there is a unique curve projected onto the 2D image plane obtained using the aforementioned camera system. Hence, if in an image, obtained through such a camera system, a curve satisfies the criteria of belonging to the family of spherical-object projection curves, it is possible to uniquely identify the spherical object's 3D world position which was projected onto that image curve. Additionally, in order to practically verify the bijection principle, we present simple experiments. Using the bijection principle, a 3D sphere detection algorithm is implemented [15], plugged in a particle filter-based tracker [15] and then applied on a soccer robot to detect and track a FIFA size-5 ball. Note that we have already applied this principle in some of our previous works [15–17] with extensive experimental results establishing its practical correctness and applicability. The experiments described and separately developed for this article is included for the sake of complete and clear understanding of the bijection principle and its utility.

1.4. Organization of the article

The rest of the article is structured as follows. In Section 2, we overview the fisheye equidistant projection model used in this work, which is essential for the understanding of the bijection principle and its proof presented in Section 3. In Section 4, we present experiments and results including a comparison with the ground truth. We conclude the article with final remarks in Section 5.

2. Fisheye lens equidistant projection model

In Fig. 1, A is a point in the 3D world frame which after being projected through the fisheye lens, falls on the point M which lies on the image plane (blue[1] color plane in Fig. 1). The projection follows a non-linear transformation function when expressed in the Cartesian coordinate system, but is linear in the spherical coordinate system (1) and is called the fisheye equidistant projection model. It is expressed as

$$d = f\Theta, \tag{1}$$

where the 3D world frame follows a spherical coordinate system with coordinate variables denoted by r, Θ and Φ and the 2D image frame follows a polar coordinate system with coordinate variables d

[1] For interpretation of color in Fig. 1, the reader is referred to the web version of this article.

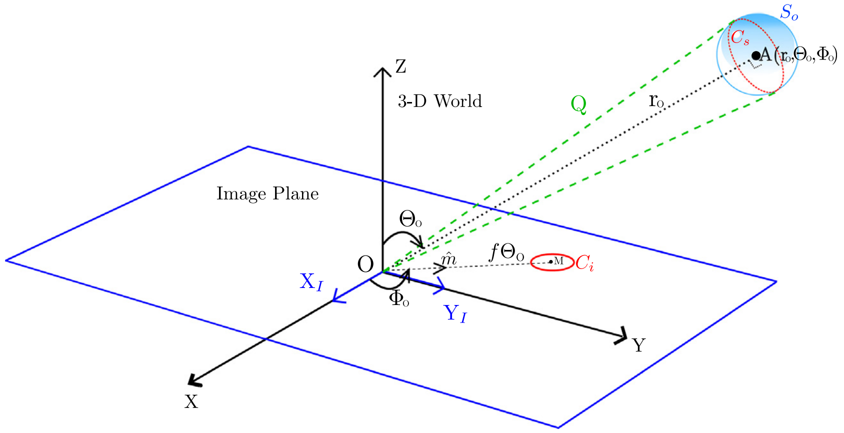


Fig. 2. 3D object to 2D image projection.

and Φ. Since Φ remains unchanged after the transformation it is denoted by the same variable. f denotes the focal length of the lens.

3. The bijection principle

In this section we propose and prove the *3D spherical object to 2D image bijection principle* which states that the periphery of a spherical object of known radius when observed through a fisheye lens following the equidistant projection model (1), always projects into a unique curve in the image frame for each possible 3D position of that object. Conversely, each curve in the image, which satisfies the condition of being projected from the periphery of a known-radius spherical object, back projects into a unique 3D position of that spherical object.

In order to prove this principle, we first introduce the coordinate reference frames for the 3D spherical object and the 2D image and the transformation from one frame to the other. We then find the expression for the curve (later referred as C_s) of the outer periphery of the sphere in 3D in the world frame, which the camera can observe. The projection model (1) is then applied to the curve C_s to obtain the projected curve C_i on the image plane in the 2D image coordinate frame. This is followed by the proof of uniqueness of the curve C_i for any 3D position of the sphere's (initial spherical object which was observed) center and vice versa hence proving the bijection principle.

3.1. Coordinates and transformations

The origin of both the image frame (polar) and the reference plane of the world frame (spherical) is at O in Fig. 1. The camera's lens which faces upwards in the positive Z axis direction is centred at the origin O in this figure. The Z axis is the zenith direction and the X–Y plane is the reference plane of the world frame as well as the plane which contains the image plane. M is the projected image point of A in the image frame, the distance of which from O is given by $d = f\Theta$ where f is the focal length of the fisheye lens and Θ (in radians) is the inclination angle, the angle which the ray joining the point $A(r,\Theta,\Phi)$ in 3D world frame and the lens' optical center (O in Fig. 1) makes with the optical axis of the lens (Z axis in Fig. 1). The azimuth angle Φ of the point A in the world frame is equal to the polar angle of its projected point M on the image plane. The equidistant projection model (1) can also be written as a transformation equation in the matrix form (2).

$$\begin{bmatrix} 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r \\ \Theta \\ \Phi \end{bmatrix} = \begin{bmatrix} d \\ \Phi \end{bmatrix}. \tag{2}$$

The rest of the equations and figures in this proof use these coordinate frames and transformations consistently. (r,Θ,Φ) are henceforth used as the spherical coordinate variables (world frame's coordinate system in which the 3D object resides) and (d,Φ) are henceforth used as the polar coordinate variables (image frame's coordinate system). Note that the variable Φ is the same in both coordinate systems because the equidistant projection model (1) keeps it unchanged and hence a common variable can be used without any loss of generality.

3.2. Projection model applied to a spherical object

In Fig. 2, a spherical object S_o of a known radius R_o is centred at the point A, the coordinates of which are (r_o,Θ_o,Φ_o) in the 3D world frame. The outer periphery of S_o visible to the camera is the circle C_s which is a circle formed by the contact of S_o's tangential cone Q from O on to S_o itself (see Fig. 2). The visible periphery C_s is a circle in 3D world frame which forms the base of the cone Q and has a radius R_1 ($R_1 < R_o$) (see Fig. 3). According to the projection model (1), each point on C_s is projected onto the image plane forming a curve C_i. In order to prove the proposed bijection principle, it is essential to find the equation of the curve C_i in terms of the polar coordinate variables (d,Φ). The sphere S_o's center's 3D coordinates r_o, Θ_o, Φ_o will serve as the parameters for C_i's curve equation. The lens' focal length f and the S_o's known radius R_o will be the fixed constants in C_i's curve equation. To do so we first find the equation of C_s in terms of the world frame's spherical coordinate

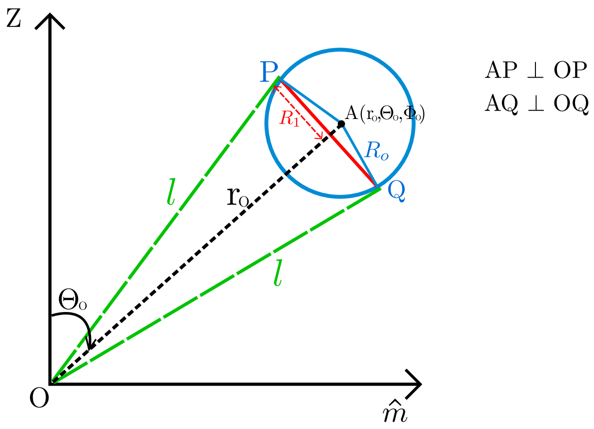


Fig. 3. Cross-section of Fig. 2 along the plane perpendicular to X–Y plane and at an azimuth of Φ_0. $\hat{m}$ is the vector along the azimuth of Φ_0 on the reference plane of the world frame in Fig. 2.

variables (r, Θ, Φ) and then apply (1) on the coordinate variables in its equation to find C_i. Next, in order to prove the bijection we will show that for a given triplet of parameters (r_o, Θ_o, Φ_o), *i.e.*, for a given position of the sphere S_o the curve equation C_i represents a unique curve on the image plane. This is followed by proving the converse, *i.e.*, if there exists a curve of the form C_i in the image plane, the coefficients of the terms in C_i's expression will uniquely represent a single point in the world frame's 3D space at which the original spherical object S_o is centred.

3.3. Derivation of the equation of curve C_s

Observing from Figs. 2 and 3, C_s is a circle which lies on the intersection of the plane PQ (observed as a line in the cross section Fig. 3) and the sphere S_o. The equation of the sphere S_o (3) in the world reference frame's spherical coordinate system is obtained by transforming a sphere's standard equation in the Cartesian coordinate system into spherical coordinate system.

$$S_o(r,\Theta,\Phi) = \{(r,\Theta,\Phi) : r^2 + r_o^2 - 2rr_o(\sin\Theta\sin\Theta_o\cos(\Phi-\Phi_o) + \cos\Theta\cos\Theta_o) - R_o^2 = 0\}. \tag{3}$$

Since the cone Q is tangential to the sphere S_o, all points on the curve C_s are at the same distance (denoted by l) from the apex O of the cone Q (see Fig. 3). As per this argument, in Eq. (3), where r is the coordinate variable denoting the distance of any point on S_o from the origin O, substituting r with the value of l will generate the equation of the curve C_s. From Fig. 3, the value of l in terms of the sphere's parameters and constants is derived as:

$$l = \sqrt{r_o^2 - R_o^2}. \tag{4}$$

Note that one must assume that the radius of the sphere R_o is less than the distance of its center to the origin r_o. Performing the substitution of r with the value of l in the equation of S_o (3), we obtain the equation of the curve C_s as follows:

$$\begin{aligned} & C_s(r,\Theta,\Phi) = S_o(r,\Theta,\Phi)|_{r=l} \\ \Rightarrow\ & C_s(r,\Theta,\Phi) = \{(l,\Theta,\Phi) : r_o^2 \\ & - r_o\sqrt{r_o^2 - R_o^2}(\sin\Theta\sin\Theta_o\cos(\Phi-\Phi_o) \\ & + \cos\Theta\cos\Theta_o) - R_o^2 = 0\}. \end{aligned} \tag{5}$$

3.4. Obtaining C_i from C_s

Applying (1) on the coordinate variables of the equation of C_s we obtain

$$C_i(d,\Phi) = \left\{(d,\Phi) : r_o^2 - r_o\sqrt{r_o^2 - R_o^2}\left(\sin\left(\frac{d}{f}\right)\sin\Theta_o\cos(\Phi-\Phi_o) + \cos\left(\frac{d}{f}\right)\cos\Theta_o\right) - R_o^2 = 0\right\} \tag{6}$$

which is expressed in the image's polar coordinate variables d and Φ, the parameters r_o, Θ_o, Φ_o which are the spherical coordinates of the sphere S_o's center and the fixed constants f (lens' focal length) and R_o (sphere S_o's known radius). We represent the curve $C_i(d,\Phi)$ as a function $\mathcal{F}(d,\Phi)$ with constraints on the variables and parameters in the following way:

$$\mathcal{F}(d,\Phi) = \left\{(d,\Phi) : r_o^2 - r_o\sqrt{r_o^2 - R_o^2}\left(\sin\left(\frac{d}{f}\right)\sin\Theta_o\cos(\Phi-\Phi_o) + \cos\left(\frac{d}{f}\right)\cos\Theta_o\right) - R_o^2 = 0; 0 < d \leqslant f\frac{\pi}{2}; -\pi \leqslant \Phi \leqslant \pi\right\}. \tag{7}$$

where $R_o,\ f > 0;\ r_o > R_o;\ 0 \leqslant \Theta_o < \frac{\pi}{2};\ -\pi \leqslant \Phi_o \leqslant \pi$ and $d,\ \Phi,\ r_o,\ \Theta_o,\ \Phi_o,\ f,\ R_o \in \mathbb{R}$.

The restricted positive limit of d in (7) is due to the fact that the image projection of any real point in the 3D world frame can attain a maximum value of $d = f\frac{\pi}{2}$ due to the projection model (1). The limits of Θ_o in (7) are due to the fact that only the points on the positive side of the Z axis can form a real image if the camera's lens is placed at the origin O and is pointing in the positive Z axis direction. The fixed constants: camera lens' focal length f and the sphere's radius R_o, represent positive valued quantities by definition. The distance r_o from the principal point of the image (which is also the origin of the spherical coordinate system) to the center of the sphere is greater than the radius of the sphere implying that the camera is always outside the sphere S_o. The rest of the limits on the variables and parameters in (7) are simply equivalent to the limits of the coordinate variables in a standard polar or spherical coordinate system.

3.5. Proof of bijection

Principle 1 (*3D spherical object to 2D image bijection principle*). *The periphery of a spherical object of known radius, when observed through a fisheye lens that follows the equidistant projection model (1), always projects into a unique curve in the image frame for each possible 3D position of that object. Conversely, each curve in the image, which satisfies the condition of being projected from the periphery of a known sized spherical object, back projects into a unique 3D position of that spherical object.*

3.5.1. Mathematical formulation

The bijection principle, as stated above, can be mathematically formulated in the following way: A map $\mathcal{G}:\ \mathbb{S} \to \mathbb{C}$ is bijective[2] where

$$\mathbb{S} = \left\{(r_o,\Theta_o,\Phi_o) : r_o > R_o, 0 \leqslant \Theta_o < \frac{\pi}{2}, -\pi \leqslant \Phi_o \leqslant \pi; r_o,\Theta_o,\Phi_o \in \mathbb{R}\right\}, \tag{8}$$

$$\mathbb{C} = \{C_i : C_i \equiv \mathcal{F}(d,\Phi)\}, \tag{9}$$

for fixed values of f and R_o.

In order to prove the bijection principle, we need to establish the following two statements:

- The map $\mathcal{G}:\ \mathbb{S} \to \mathbb{C}$ is injective.
- The map $\mathcal{G}:\ \mathbb{S} \to \mathbb{C}$ is surjective.

3.5.2. Proof of injection

Simplifying (7) we get:

$$\mathcal{F}(d,\Phi) = \left\{(d,\Phi) : 1 - \alpha\sin\left(\frac{d}{f}\right)\cos(\Phi) - \beta\sin\left(\frac{d}{f}\right)\sin(\Phi) - \gamma\cos\left(\frac{d}{f}\right) = 0\right\}, \tag{10}$$

where

$$\begin{aligned} \alpha &= \frac{r_o\sin\Theta_o\cos\Phi_o}{\sqrt{r_o^2 - R_o^2}}; \\ \beta &= \frac{r_o\sin\Theta_o\sin\Phi_o}{\sqrt{r_o^2 - R_o^2}}; \\ \gamma &= \frac{r_o\cos\Theta_o}{\sqrt{r_o^2 - R_o^2}}, \end{aligned} \tag{11}$$

assuming the restrictions on the variables and the parameters to be the same as in (7).

[2] Note that here and later in this article the symbol $\mathbb{C}$ is not a notation for the complex plane. It only represents the set as defined in (9).

In order to prove that the map $\mathcal{G}$ is injective, we need to show that any given triplet (r_o, Θ_o, Φ_o), for all $r_o > R_o$, $0 \leqslant \Theta_o \leqslant \frac{\pi}{2}$, $-\pi \leqslant \Phi_o \leqslant \pi$, maps to a unique curve of the form C_i which implies that the mapping is one to one. Since a curve is a set of points, it implies that we need to show that a given triplet (r_o, Θ_o, Φ_o) maps to a unique set of points in the polar space of (d, Φ) for all $0 < d \leqslant f\frac{\pi}{2}$; $-\pi \leqslant \Phi \leqslant \pi$.

The forward mapping of $\mathcal{G}$ is trivial. Considering the simplified Eq. (10), we can find a set $\mathcal{K}$ of all the points in the polar space (d, Φ) which satisfy (10) for a given triplet (r_o, Θ_o, Φ_o). A given triplet (r_o, Θ_o, Φ_o) implies that the values of the coefficients α, β and γ are given.

The mapping $\mathcal{G}$ is one to one, *i.e.*, injective if and only if the set of points $\mathcal{K}$ maps back to the unique triplet (r_o, Θ_o, Φ_o). For the inverse mapping of $\mathcal{G}$, assume that we are given the set of points $\mathcal{K}$ such that all the points $(d, \phi) \in \mathcal{K}$ satisfy the equation of the form (10). Let us pick three distinct points $((d_1, \Phi_1), (d_2, \Phi_2), (d_3, \Phi_3)) \in \mathcal{K}$. In order to show that three distinct points exist on the curve (10), we will first prove that at least four distinct points exist on (10).

Notice that in (10), in the family of equations represented, the function equated to 0 can be seen in two forms; either Φ as a function of d or d as a function of Φ. By differentiating such representations of this function w.r.t. d in the first form and w.r.t Φ in the other and then separately equating both differentials to zero, we can obtain the values for d and Φ at which the function has the maxima and minima. Eqs. (12)–(15) represent these extrema.

$$d_{max} = f\left(\Theta_o + \arcsin\frac{R_o}{r_o}\right) \tag{12}$$

$$d_{min} = f\left(\Theta_o - \arcsin\frac{R_o}{r_o}\right) \tag{13}$$

$$\Phi_{max} = \Phi_o + \arcsin\frac{R_o}{r_o \sin\Theta_o} \tag{14}$$

$$\Phi_{min} = \Phi_o - \arcsin\frac{R_o}{r_o \sin\Theta_o} \tag{15}$$

At the extrema of d, the corresponding value of Φ and at the extrema of Φ, the corresponding value of d can be obtained by plugging the values of the extrema back into (10). These points in polar coordinates will be given by $\left(\left(\Theta_o + \arcsin\frac{R_o}{r_o}\right), \Phi_o\right)$, $\left(f\left(\Theta_o - \arcsin\frac{R_o}{r_o}\right), \Phi_0\right)$, $\left(f\left(\arccos\frac{r_o\cos\Theta_o}{\sqrt{r_o^2 - R_o^2}}\right), \Phi_o + \arcsin\frac{R_o}{r_o\sin\Theta_o}\right)$ and $\left(f\left(\arccos\frac{r_o\cos\Theta_o}{\sqrt{r_o^2 - R_o^2}}\right), \Phi_o - \arcsin\frac{R_o}{r_o\sin\Theta_o}\right)$.

It can now be verified that these represent four distinct points assuming the constraints on the constants of (10) explained earlier in this section.

Proceeding with the proof of injection, since by initial assumption all the points in the set $\mathcal{K}$ satisfy the curve Eq. (10), the following three equations are true:

$$\alpha \sin\left(\frac{d_1}{f}\right)\cos(\Phi_1) + \beta \sin\left(\frac{d_1}{f}\right)\sin(\Phi_1) + \gamma \cos\left(\frac{d_1}{f}\right) = 1 \tag{16}$$

$$\alpha \sin\left(\frac{d_2}{f}\right)\cos(\Phi_2) + \beta \sin\left(\frac{d_2}{f}\right)\sin(\Phi_2) + \gamma \cos\left(\frac{d_2}{f}\right) = 1, \tag{17}$$

$$\alpha \sin\left(\frac{d_3}{f}\right)\cos(\Phi_3) + \beta \sin\left(\frac{d_3}{f}\right)\sin(\Phi_3) + \gamma \cos\left(\frac{d_3}{f}\right) = 1. \tag{18}$$

In the case of inverse mapping of $\mathcal{G}$, $\mathcal{K}$ is given and hence known but the triplet (r_o, Θ_o, Φ_o) is unknown, hence α, β, γ become variables. Eqs. (16)–(18) can now be seen as a system of 3 equations in 3 variables: α, β, γ. Expressing this system in matrix form:

$$\mathcal{A}\rho = 1_3, \tag{19}$$

where

$$\mathcal{A} = \begin{bmatrix} \sin\left(\frac{d_1}{f}\right)\cos(\Phi_1) & \sin\left(\frac{d_1}{f}\right)\sin(\Phi_1) & \cos\left(\frac{d_1}{f}\right) \\ \sin\left(\frac{d_2}{f}\right)\cos(\Phi_2) & \sin\left(\frac{d_2}{f}\right)\sin(\Phi_2) & \cos\left(\frac{d_2}{f}\right) \\ \sin\left(\frac{d_3}{f}\right)\cos(\Phi_3) & \sin\left(\frac{d_3}{f}\right)\sin(\Phi_3) & \cos\left(\frac{d_3}{f}\right) \end{bmatrix}, \tag{20}$$

$\rho = [\alpha \quad \beta \quad \gamma]^T$ and $1_3 = [1 \quad 1 \quad 1]^T$.

The system in (19) has a unique solution for ρ if and only if $\mathcal{A}$ is invertible. Also, from (11) it can be easily verified that the triplet (r_o, Θ_o, Φ_0) can be uniquely obtained from the triplet (α, β, γ). Hence in order to show that the map $\mathcal{G}^{-1}$ will map from a set $\mathcal{K}$ to a unique triplet (r_o, Θ_o, Φ_0) and therefore prove that the map $\mathcal{G}$ is injective, it will suffice to show that $\mathcal{A}$ is invertible under the initial assumptions of the problem.

$$\mathcal{A}^{-1} = \begin{bmatrix} \dfrac{\frac{\sin\left(\frac{d_3}{f}\right)\sin(\phi_1)}{\tan\left(\frac{d_2}{f}\right)} - \cos\left(\frac{d_3}{f}\right)\sin(\Phi_2)}{\sin\left(\frac{d_1 - d_3}{f}\right)\sin(\Phi_1 - \Phi_2)} & \dfrac{\sin(\phi_1)}{\sin\left(\frac{d_2}{f}\right)\sin(\Phi_1 - \Phi_2)} & \dfrac{-\frac{\sin\left(\frac{d_1}{f}\right)\sin(\phi_1)}{\tan\left(\frac{d_2}{f}\right)} + \cos\left(\frac{d_1}{f}\right)\sin(\Phi_2)}{\sin\left(\frac{d_1 - d_3}{f}\right)\sin(\Phi_1 - \Phi_2)} \\ \dfrac{\frac{-\sin\left(\frac{d_3}{f}\right)\cos(\phi_1)}{\tan\left(\frac{d_2}{f}\right)} + \cos\left(\frac{d_3}{f}\right)\cos(\Phi_2)}{\sin\left(\frac{d_1 - d_3}{f}\right)\sin(\Phi_1 - \Phi_2)} & -\dfrac{\cos(\phi_1)}{\sin\left(\frac{d_2}{f}\right)\sin(\Phi_1 - \Phi_2)} & \dfrac{\frac{\sin\left(\frac{d_1}{f}\right)\cos(\phi_1)}{\tan\left(\frac{d_2}{f}\right)} - \cos\left(\frac{d_1}{f}\right)\cos(\Phi_2)}{\sin\left(\frac{d_1 - d_3}{f}\right)\sin(\Phi_1 - \Phi_2)} \\ -\dfrac{\sin\left(\frac{d_3}{f}\right)}{\sin\left(\frac{d_1 - d_3}{f}\right)} & 0 & \dfrac{\sin\left(\frac{d_1}{f}\right)}{\sin\left(\frac{d_1 - d_3}{f}\right)} \end{bmatrix}. \tag{21}$$

In the expression of $\mathcal{A}^{-1}$ (21), the inverse of matrix $\mathcal{A}$, it can be verified that under the previously mentioned variables and constants limits, all elements of $\mathcal{A}^{-1}$ will have a finite value thus proving its existence. Hence the map $\mathcal{G}$ is injective.

3.5.3. Proof of surjection

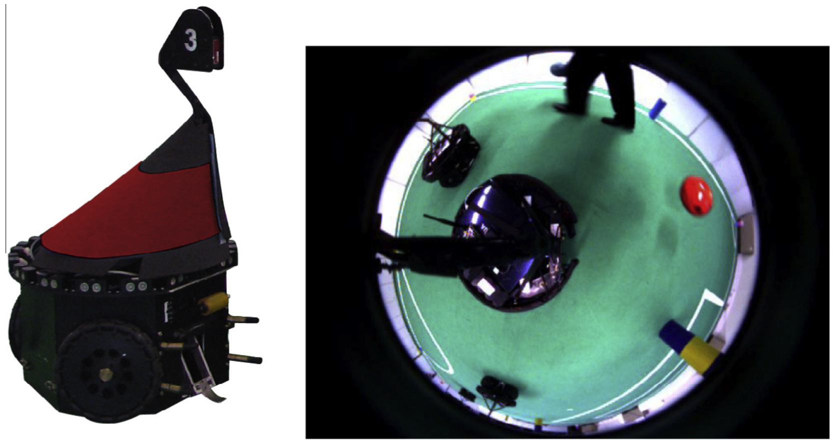
The proof of surjection is now trivial. A function is surjective if its image[3] is equal to its codomain. We have already shown that any set of points $\mathcal{K} = \{(d, \Phi)\}$ in the polar coordinate space which satisfies the equation of the form (10) (*i.e.*, $\mathcal{K}$ is in the codomain $\mathbb{C}$ of the mapping $\mathcal{G}$) maps back to a unique triplet $(r_o, \Theta_o, \Phi_o) \in \mathbb{S}$, implying that $\mathcal{K}$ is also in the image of $\mathcal{G}$. Therefore, the codomain of the map $\mathcal{G}$ is equal to its image. Hence the map is surjective.

A mapping is bijective if and only if it is injective and surjective at the same time. By establishing that the map $\mathcal{G}$ is injective and surjective, we conclude that it is bijective. Hence, proving the proposed bijection principle. □

3.6. Inference

Concluding from all the previous subsections in this section, it is possible to uniquely identify the position of a known-radius spherical object in 3D using a single fisheye equidistant projected image. Using this bijection principle, in [15] we described a fast algorithm to (i) detect the curves of the form C_i (referred in [15] as tear-drop

[3] Not to be confused with the camera image mentioned elsewhere in this article. In this paragraph image refers to the image (also called 'range') of a mathematical function/mapping.



(a) MSL team SocRob's omni-directional robot.

(b) Example image from the robot's on-board dioptric vision system

Fig. 4. Side view of the robot used in the experiments and an example image acquired from the robot's fisheye lens-based camera, with an orange soccer ball, landmarks and other robots in the field of view. The bijection principle is applied in real-time on such images which are continuously acquired by the robot's camera. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

curves because of their visual appearance) in the image and, (ii) find the parameters of those curves. Subsequently, the 3D position of the center of the spherical object could be found using those parameters in a straightforward manner.

The detected position can then be used for various next-level tasks based on the required application, e.g., tracking the spherical object, recovering the trajectory of the object and thus experimentally finding out the motion model of the object, etc. In this paper, we utilize the position detection to perform spherical object tracking.

4. Experiments and results

In order to highlight the bijection principle's practical correctness, applicability and usability we present results from two separate real-robot experiments. These experiments were done using the same soccer robot tracking a FIFA size-5 ball (which is a spherical object of known radius) in two different environments. Here, different environments mean separate robotics laboratories with different lighting conditions, density of occlusions between the robot and the ball within the environment, etc. While the first environment was situated at the LSA lab facility of ISEP, Porto, the second was at Mobile Robotics Laboratory (LRM) at ISR, Lisbon. For both the experiments described further, a 3D sphere detector was implemented using the optimized model-fitting approach [15], which in turn uses the bijection principle. A particle filter-based (PF) tracker was constructed involving the standard PF's predictor and resampler [18] along with the model-fitting approach as the classifier. The model-fitting approach-based 3D sphere detector is used to assign weights to the particles in the update step of the PF.

One of the primary requisites for any robot to be able to play soccer is to continuously track the soccer ball. In robotic soccer's dynamic environment the ball maneuvers in a 3D space, which makes the robot soccer scenario an interesting and suitable testbed for the experimental validation and performance check of the bijection principle. After a brief overview of the robot used in these experiments and the robot's vision system, we describe the ground truth system used to compute the accuracy of the tracking. This is followed by an analysis of results of both the experiments. Related to these experiments a self-explanatory evidence Video is also attached with this article.

4.1. Robot and its vision system

The robot used here, as depicted in Fig. 4a, is from the RoboCup Middle Sized League (MSL) team SocRob. It is characterized by a 3-wheeled omni-directional drive and a dioptric vision system consisting of a fisheye lens-based camera facing downwards. The robot is also equipped with a laptop (Quad Core Intel (R) Core (TM) i3 CPU M350 @ 2.27 GHz, 2.9 GB RAM), which runs the 3D detection and tracking in realtime.

The robot's vision system is based on an AVT Marlin F-033C firewire camera, which is equipped with a fisheye lens providing a field-of-view of 185°, facing downwards. This dioptric system endows the robot with omnidirectional vision, capable of detecting relevant objects (e.g., the ball and other robots) at a distance of up to 3.5 m in all directions. An example image from this vision system is presented in Fig. 4b.

According to the official rules of the MSL, the robots cannot exceed a height of 80 cm above the ground plane. Accommodating with this rule, in addition to the fact that we place the camera facing downwards (to have the full field view at once), and considering other components associated with our camera system, the camera placement on the robot is such that it can detect objects up to a height of ~ 70 cm above the ground plane. From a theoretical standpoint (concerning the proposed bijection principle), the position of the camera w.r.t. the ground plane does not concern the principle itself. Making the camera position higher would only change the distance of the detected object from the camera and therefore the area of the projected blob on the image plane. It will not affect the 'family of curves' (referred to as tear-drop curves in the Section 3) that defines those projected blobs based on the bijection principle. Moreover, by keeping the same camera placement as described above and not increasing its height, we ensure fairness when comparing our work with other similar works from the MSL community [1,10,13].

4.2. Ground truth system

In order to evaluate the implementation results of our approach, we compare the estimated positions of the orange ball with its ground truth (GT). To obtain the GT, an overhead stereo vision system was used which estimates the exact 2D world positions of the robot and the 3D world positions of the soccer ball

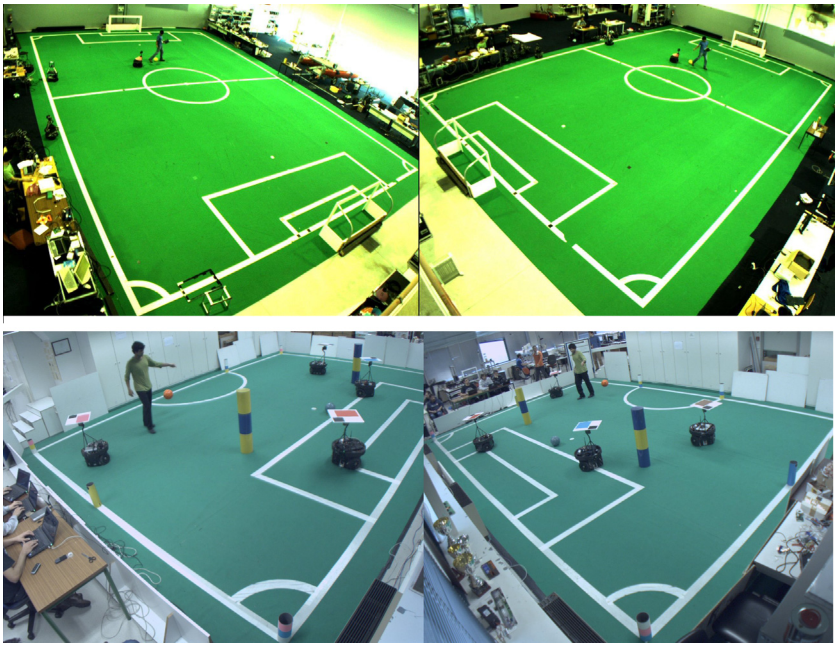


Fig. 5. Snapshot from the stereo vision system installed for ground-truth estimation of the robot's 2D and the ball's 3D positions. LSA lab facility of ISEP, Porto, Portugal (Top). Mobile Robotics Laboratory (LRM) at ISR, Lisbon, Portugal (Bottom).

which the robot in our testbed is detecting and tracking. The robot and the GT system were time-synchronized using the open-source software named Chrony.[4] Chrony was installed on all the robot laptops and the computer connected to the GT system. Moreover, all robot laptops and the GT system's computer were on the same local area network. Chrony synchronizes the clocks of all the systems on the same network on which it is installed. It implements the network time protocol (NTP) and guarantees clock synchronization accuracy in the order of microseconds.

Our GT system is based on the concepts presented in [19]. The hardware and software implementation of our GT system was done separately from the one used in [19]. It consists of 2 gigabit ethernet cameras in a stereo baseline. The GT system's cameras were positioned looking towards the testbed with a baseline of ~13 m in LSA, Porto, and ~12 m in LRM, Lisbon. They are connected to a machine with Quad Core Intel (R) Core (TM) i5 CPU 750 @ 2.67 GHz, 8 GB RAM, running a Linux operating system. The model of the camera is Basler acA1300–30gc with a maximum limit of ~25 frames per second (fps) and a resolution of 1294 × 964 pixels (1.2 megapixels). Both environments have the same GT system (meaning, the camera make and model) except for the different lengths of the baseline. The top row in Fig. 5 shows the right and left camera images from the GT system installed at LSA, Porto, while the bottom row images are from the GT system at LRM, Lisbon.

Before we set out to use the above described GT system to evaluate the performance of our proposed methods (presented in the later subsection), it becomes essential to explore and characterize the errors within the GT system itself. As the GT system employs a color-based blob-detection technique to locate the ball's/robot's position in the stereo camera images, it is natural that errors in color segmentation due to different lighting throughout the field, noise in blob detection, etc., will corrupt the GT values of the ball's/robot's position. In order to characterize this error in the GT system we followed the following procedure.

At 40 known locations on the field (see Fig. 6), five 1-meter intervals in the *X*-axis and eight 1-meter intervals in *Y*-axis, small white markers were fixed. On each of these locations the ball was kept at 3 different heights (*Z*-axis of the field). First, directly on the field surface, putting the ball's center at 0.10 m in the positive *Z*-axis direction, while the rest two on known-height stands of 0.35 m and 0.5 m, as depicted in the second and third images of Fig. 6. Each of these 120 placements of the ball was kept for approximately 1 min. During each of those 1 min intervals, images from the stereo cameras of the GT system were saved at ~ 20 Hz in order to compute the GT *X*, *Y*, *Z* position values of the ball. Error in each dimension of GT is then calculated as the absolute value of the difference between (i) the GT estimated ball position in that particular dimension and (ii) actual[5] known value of the ball position in that dimension. Subsequently, mean and variance of the error in the GT *X*, *Y*, *Z* position estimates for each of the 120 placements was obtained.

In order to describe how the error in GT estimation varies over each of the three coordinate dimensions of the field, we present a set of plots in Fig. 7. For each plot, we fix one of the coordinates and average the mean error over all the ball placements (in the other two dimensions) at that fixed coordinate. For example, to describe how the *X* component of the GT varies over the *Y*-axis of the field, at each *Y* coordinate of the field we obtain the average of the mean errors in GT's *X* component for all ball placements at that *Y* coordinate. Overall, we observe that the errors in the GT estimation of the ball does not vary significantly over the field in any of the dimensions. The mean error in all dimensions remains between 0.01 m and 0.04 m with an average close to 0.02 m indicating a good accuracy for 1.2 megapixel cameras observing the ball up to 9 m away from the center of the stereo baseline, while the variance of error remains extremely low (in the order of 10^{-5} m^2 to 10^{-6} m^2) indicating high precision in the GT estimation. In the evidence Video attached with this article, one can also visually verify the precision and accuracy of the GT system. Considering the fact that the GT estimates for the robots were made using an exact same method as that of the ball (by placing colored plates on top of the robots and employing similar color-based blob-detection technique), we

[4] Webpage of the software Chrony: http://chrony.tuxfamily.org/index.html.

[5] Here, actual values refer to the 40 fixed placements of the white markers and the 3 separate fixed heights as explained in the paragraph.

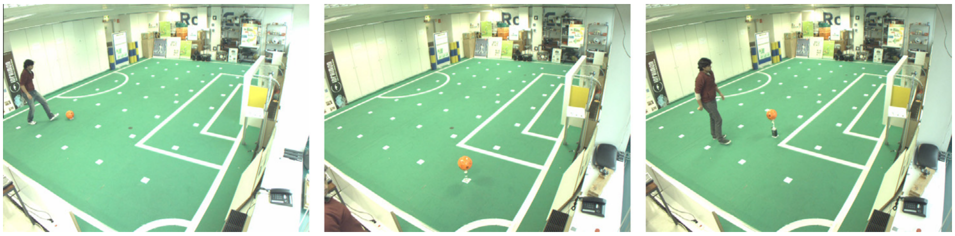
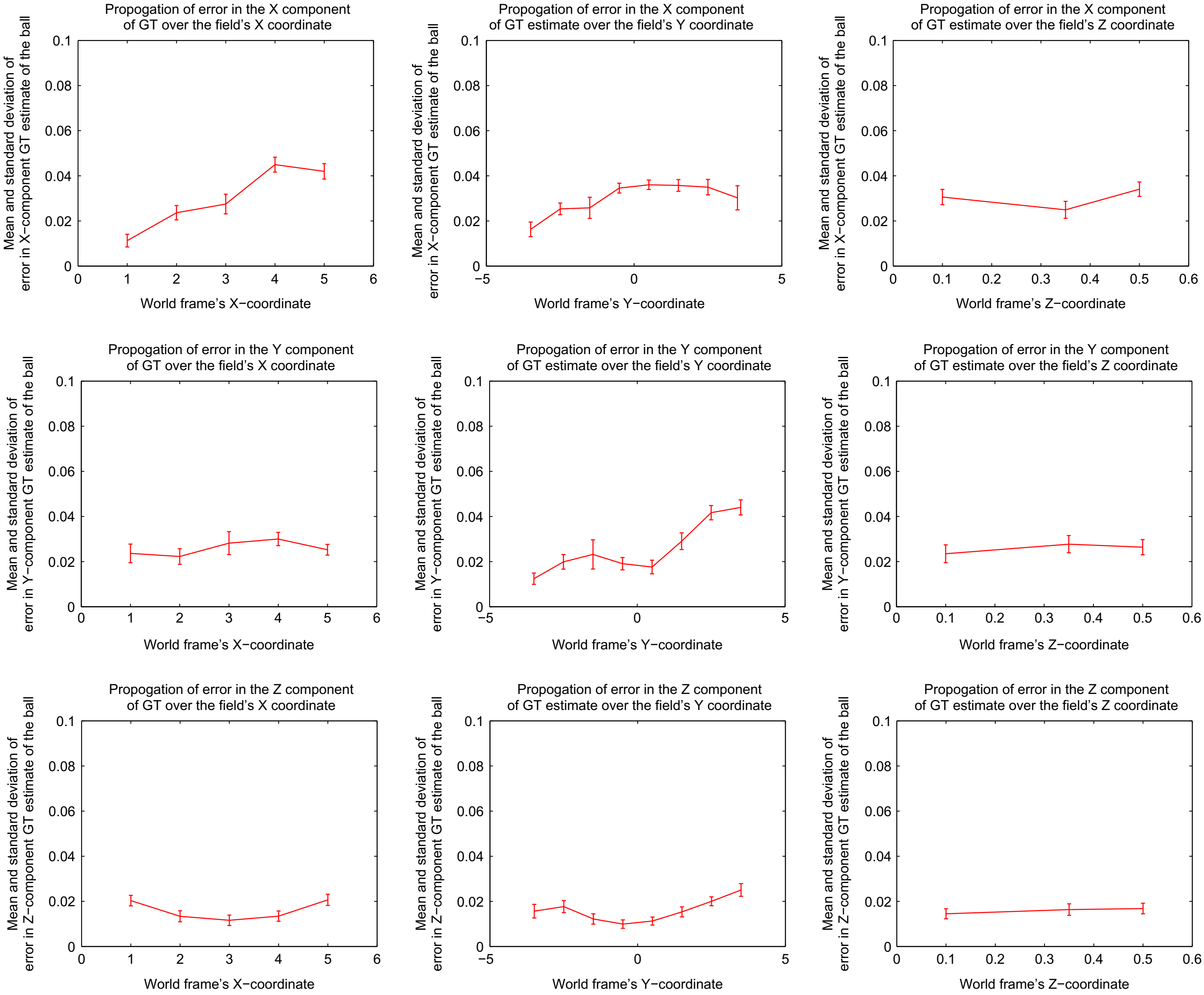


Fig. 6. Characterization of error within the ground truth (GT) estimation of the ball. Each of the above images show a different *X*, *Y* and *Z* coordinate placement of the ball from the right camera of the GT system at ISR, Lisbon. The origin of world-frame coordinates is at the center of the soccer field with positive *X* axis towards the yellow colored goal. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



Propogation of error in the X component of GT over the field's X coordinate

Propogation of error in the X component of GT estimate over the field's Y coordinate

Propogation of error in the X component of GT estimate over the field's Z coordinate

Propogation of error in the Y component of GT over the field's X coordinate

Propogation of error in the Y component of GT estimate over the field's Y coordinate

Propogation of error in the Y component of GT estimate over the field's Z coordinate

Propogation of error in the Z component of GT over the field's X coordinate

Propogation of error in the Z component of GT estimate over the field's Y coordinate

Propogation of error in the Z component of GT estimate over the field's Z coordinate

Mean and standard deviation of error in X−component GT estimate of the ball
Mean and standard deviation of error in Y−component GT estimate of the ball
Mean and standard deviation of error in Z−component GT estimate of the ball
0
0.02
0.04
0.06
0.08
0.1
0 1 2 3 4 5 6
−5 0 5
0 0.1 0.2 0.3 0.4 0.5 0.6
World frame's X−coordinate
World frame's Y−coordinate
World frame's Z−coordinate

Fig. 7. Mean and standard deviation of the errors in the GT estimates of the ball position. The plots describe how the errors in *X*, *Y* and *Z* components of the GT vary over each of the field dimensions. Units for both *X* and *Y* axis in the above plots is meter.

can safely assume a similar accuracy and precision in the GT estimation of the robot positions.

4.3. Results

In both of the experiments presented in this subsection, the PF-based tracker runs on the robot and estimates the 3D position of the ball in its local frame of reference. Subsequently, the robot uses its own pose (2D position and orientation) estimate in the world-frame to transform the ball's position estimate from its local frame to the world frame. This transformation is essential for the error calculation of the ball position estimates w.r.t. its GT due the following two reasons:

- The GT system provides the world frame GT of the ball's 3D positions.

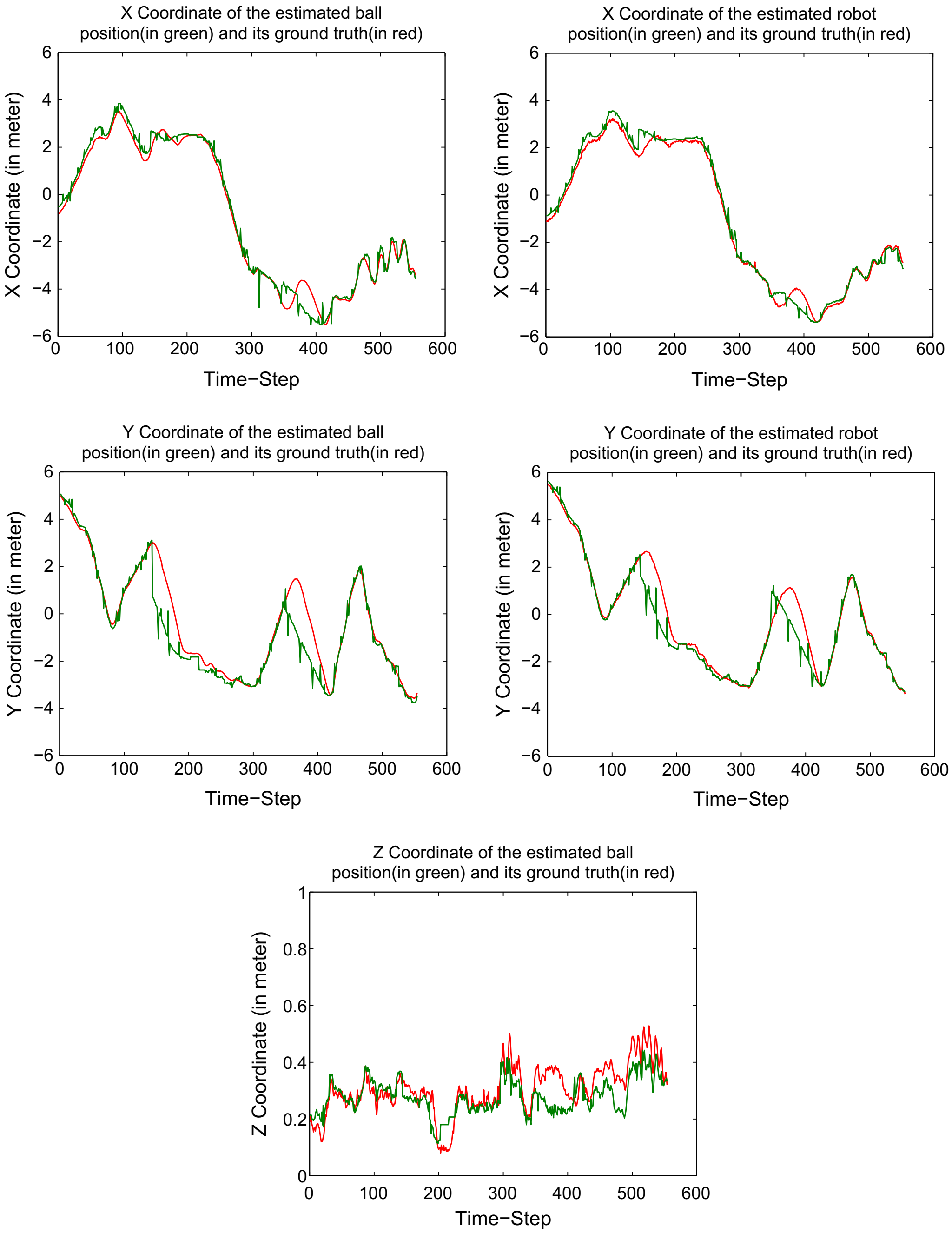


Fig. 8. Experiment 1 (in LSA, Porto): World frame trajectory plots (individual plots for each dimension: *X*, *Y* and *Z* dimensions for the ball; *X* and *Y* for the robot) of the estimated values by the robot plotted in green. Corresponding ground truth trajectories plotted in red. In all of the above plots, each time-step in the *X*-axis corresponds to ~ 0.1 s. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- The GT system provides only the 2D position GT of the robot and not the orientation GT. Hence, geometrically, it is impossible to compute the GT of the ball's local frame position estimates (in local frame's *X* and *Y* dimensions, separately).

For estimating its own pose, the robot runs a Monte Carlo self-localization algorithm, separately from the PF-based object tracker. As this self-localization is also prone to errors, when the ball position is transformed from local frame to world frame using the robot's pose, the self-localization error gets combined with the PF-based object tracking error in the final world frame ball position estimates. Note that the local *Z* coordinate position estimate of the ball is independent of the robot's pose because the robot maneuvers on fixed *X*–*Y* plane. Consequently, the local and world *Z* coordinates of the ball's position estimates are the same.

On the other hand, if one considers the radial distance estimation of the ball from the robot, the orientation of the robot itself becomes irrelevant. Therefore, given that the GT system is restricted to only 3D position GTs of the ball and the 2D position GTs of the robot, the GT of the radial distance between the robot and the ball can be easily obtained. Using this additional GT, we also compare the radial distance estimation of the ball from the robot.

For both experiments we present world frame trajectory plots (Figs. 8 and 9) for the robot and the ball in each coordinate axis (*X*, *Y* and *Z* for the ball; *X* and *Y* for the robot). On each plot, the estimates provided by the robot of its own pose and of the ball's

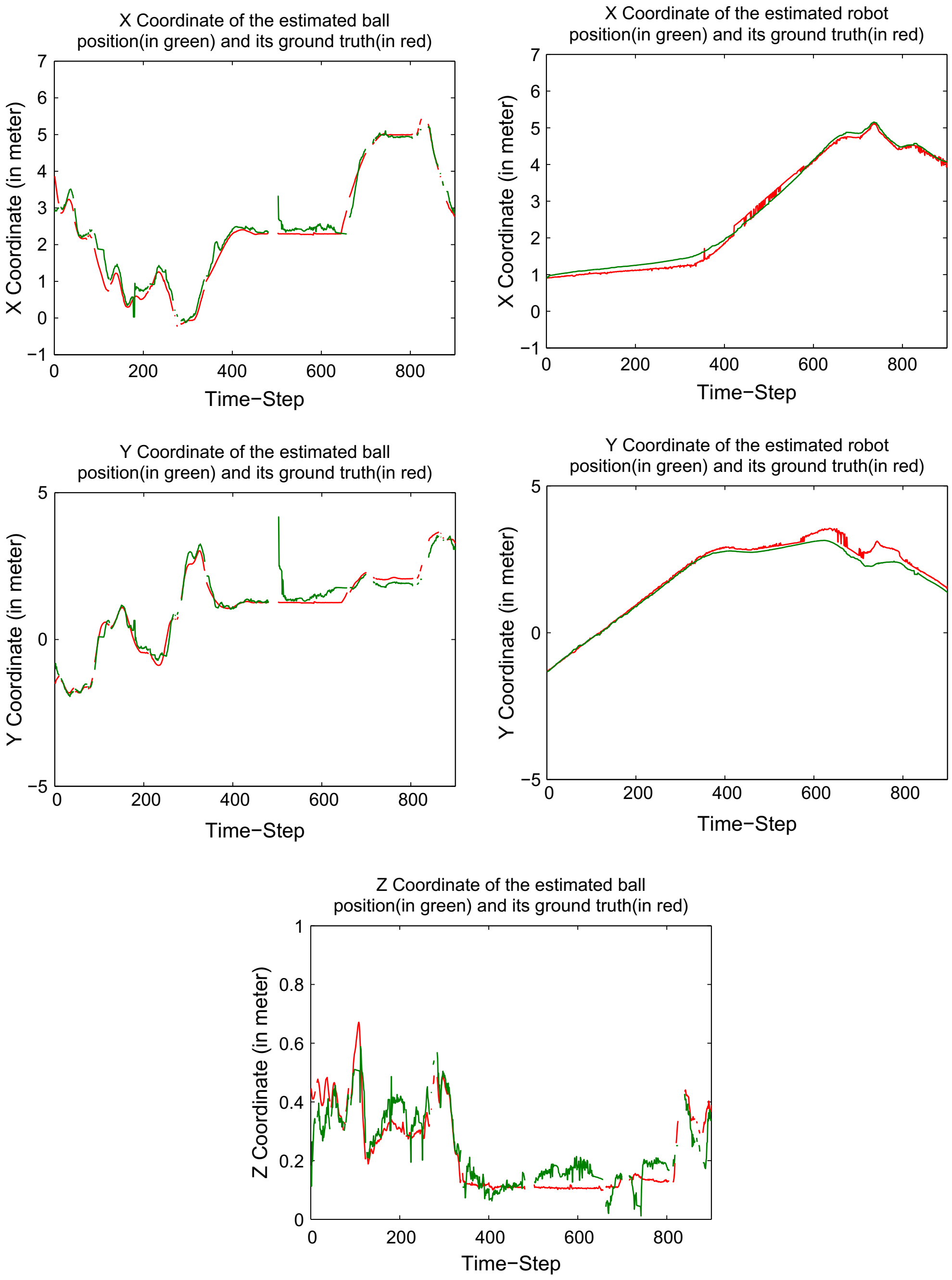


Fig. 9. Experiment 2 (in LRM, Lisbon): World frame trajectory plots (individual plots for each dimension: *X*, *Y* and *Z* dimensions for the ball; *X* and *Y* for the robot) of the estimated values by the robot plotted in green. Corresponding ground truth trajectories plotted in red. In all of the above plots, each time-step in the *X*-axis corresponds to ~0.05 s. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

world frame position is plotted in green color against their corresponding GT in red color. Error in each coordinate axis is computed as the absolute value of the difference between the value estimated by the robot and its corresponding GT. Error in radial position estimate of the ball is calculated similarly: absolute value of the difference between the following two items:

- Euclidean distance between (i) the estimated *X*–*Y* position of the robot through its own self-localization mechanism and (ii) the estimated world frame *X*–*Y* position of the ball through the PF-based tracker running on the robot, and
- Euclidean distance between (i) the *X*–*Y* GT position of the robot (ii) the *X*–*Y* GT position of the ball.

The mean and variance of all the above described errors for the full duration of each experiment is presented in Tables 1 and 2.

Experiment 1: In the first experiment the robot tracks an orange-colored ball and autonomously moves towards it while the ball is manually moved around in the 3D space around the robot on an MSL field. This experiment was conducted in the LSA lab facility of ISEP, Porto. In this experiment only a single robot was present and moving in the field, in addition to the person who was manually moving the ball tied to a thin string. A snapshot of this environment is presented in the top row images of Fig. 5. Fig. 8 and Table 1 present the results of this experiment.

Table 1
Experiment 1 (in LSA, Porto): Mean and variance of the errors calculated w.r.t. the corresponding GT values.

Errors in the World Frame						
	Ball X	Ball Y	Ball Z		Robot X	Robot Y
mean error (m)	0.267	0.459	0.049		0.226	0.355
variance of error (m^2)	0.077	0.499	0.001		0.042	0.289

Errors in the Local Frame of the Robot	
	Radial distance from Robot to Ball
mean error (m)	0.092
variance of error (m^2)	0.013

Table 2
Experiment 2 (in LRM, Lisbon): Mean and variance of the errors calculated w.r.t. the corresponding GT values.

Errors in the World Frame						
	Ball X	Ball Y	Ball Z		Robot X	Robot Y
mean error (m)	0.167	0.179	0.049		0.114	0.156
variance of error (m^2)	0.024	0.046	0.001		0.003	0.025

Errors in the Local Frame of the Robot	
	Radial distance from Robot to Ball
mean error (m)	0.174
variance of error (m^2)	0.029

During this experiment, both the robot's self-localization and the PF-based object tracker ran at approximately 30 Hz, therefore producing ball's global position estimates at the same frequency. However, the GT system captured images at a frequency of only 10 Hz. Therefore, error computation was made only at this lower frequency. Consequently, *X*-axis time-step in all the trajectory comparison plots of Fig. 8 correspond to approximately 0.1 s. Note that these values are slightly different for the second experiment.

From the mean values of the error estimates, it is clear that an accuracy of approximately 0.05 m is achieved in the *Z*-dimension of the ball's position estimate by the PF-based tracker. In the *X* and *Y* dimensions of the ball's world frame position estimates, there is a higher mean error. Simultaneously, it must be noted that the error's in the *X* and *Y* dimensions of the robot's self-localization are high as well, but less than that of the ball's world frame position errors in *X* and *Y* dimensions. This clearly suggests the additive influence of the robot's self-localization errors on the world frame ball position estimation. Although the local frame ball position error is impossible to calculate in the *X* and *Y* dimensions of the robot considering our current GT system, mean of the radial distance error estimate ($\sim$ 0.09 m) does suggest that in local frame the estimated ball position's mean error in *X* and *Y* dimensions of the robot should be less than $\sim$ 0.09 m. This, eventually, is close to the accuracy obtained in the *Z* coordinate estimates of the ball.

Experiment 2: In the second experiment, 4 similar robots are randomly moved around in an MSL field through remote control. The reason for having 4 robots is to create dynamic obstacles in the field increasing the chances of occlusions for the one robot that tracks the ball. Additionally, a person carries around an orange-colored ball (and also another blue-colored ball which is not analyzed in this experiment) tied to a string. One of four robots run the PF-based tracker (which in turn is using the bijection principle) and the MCL-based self-localization in an offline[6] manner. Another key difference in this experiment, as compared to the previous one, is in the MCL-based localization. Here, several differently colored and known landmarks are put in the environment which facilitate a more accurate self-localization. In the previous experiment, MSL field lines were used for this purpose, which, given the field's large size and robot camera's low resolution, leads to a poorer localization. This experiment was conducted in the Mobile Robotics Laboratory (LRM) at ISR, Lisbon. A snapshot of this environment is presented in the bottom row images of Fig. 5. Fig. 9 and Table 2 present the results of this experiment.

During this experiment, both the robot's self-localization and the PF-based object tracker ran at approximately 21 Hz. The GT system also captured images at a similar frequency. Therefore, it was possible to compute the errors at approximately 21 Hz. Due to the same reason, *X*-axis time-step in all the trajectory comparison plots of Fig. 9 correspond to approximately 0.05 s.

The effects of better self-localization of the robot is evident in the results of this experiments. The mean error of the robot's self-localization in the *X* and *Y* dimensions have approximately reduced to one half of what was achieved in the previous experiment. Consequently, the mean error in the *X* and *Y* dimensions of the ball's world frame position estimation have reduced by the same order. The accuracy in the *Z* dimension of the ball position estimation (mean error of approximately 0.05 m) remains the same as that of experiment 1. This was expected because the robot's self-localization is irrelevant for the *Z*-coordinate position estimation of the ball. It also confirms the fact that the PF-based tracker behaved consistently through different environmental setups. However, the mean error in the radial distance estimate of the ball from the robot has almost doubled (0.17 m) in experiment 2 as compared to that of experiment 1 (0.09 m). The reason is as follows. It is expected that as the spherical object goes further away from the robot's camera, the number of pixels projected on the image due that object rapidly decreases causing a higher inaccuracy in the overall process of object detection. In experiment 2, the ball was often carried away as far as 3–4 m from the robot. On the other hand, in experiment 1 the ball was always quite close to the robot itself (on an average, it was within $\sim$ 1 m of the robot's center). In addition to the evidence Video, this is also clearly visible from the trajectory plots. In the plots of experiment 1 (Fig. 8), it can be observed that the robot and the ball's *X*, *Y* trajectories (in the right and left columns, respectively) are very close to each other. In experiment 2 (Fig. 9), however, these trajectories are completely disparate.

5. Conclusions and future work

In this article we proposed a bijection principle which states that the 3D position of a spherical object of known radius, when projected onto an image through a fisheye lens-based camera following equidistant projection model, can be uniquely determined

[6] Offline means that the robot initially collects the raw odometry measurements and images while it is moving physically and post-processes these measurements later in a realtime scenario. For a more detailed description, please refer the experimental results section in [15], where the same concept of offline processing is used.

using only a single image frame. We presented a detailed mathematical proof for this principle which is the central contribution of this work. In order to verify the bijection principle's correctness as well as applicability in detecting spherical objects in 3D, we presented experiments to detect and track a soccer ball of known radius through a robot consisting of a single, fisheye lens-based camera. The experiments were performed under different environmental conditions, e.g., lighting conditions, occlusion densities, etc. Results highlight the accuracy and precision of tracking, which in turn experimentally verify the bijection principle and signify its practical utility. On the other hand, it must be noted that the errors in tracking occurred due to various uncontrollable factors, such as, low resolution of the camera (caused reduced visibility of the object when it is moved far from the camera), and imperfect color segmentation method involved within the detection process. These errors are not related to the bijection principle itself, which is a purely mathematical concept. Moreover, the experiments presented in this paper were in addition to the ones in our previous works where we have successfully tested detection and tracking of spherical objects in 3D using this bijection principle.

Our ongoing and future work w.r.t. the detection techniques in 3D include algorithms to detect generic objects with more complex 3D geometry than that of spheres. Applying a similar idea of bijection to objects than need to be described in a higher dimensional space is one of the directions which we intend to explore. Another aspect that we are currently investigating involves using this bijection principle to study the change in the object's projection geometry during motion blur. Furthermore, we also intend to explore the possibility of characterizing motion models of randomly moving spherical targets using the 3D sphere detection.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.cviu.2014.04.004.

References

[1] A.J. Neves, A.J. Pinho, D.A. Martins, B. Cunha, An efficient omnidirectional vision system for soccer robots: from calibration to object detection, Mechatronics 21 (2011) 399–410 (Special Issue on Advances in intelligent robot design for the Robocup Middle Size League).

[2] D. Schneider, E. Schwalbe, H.-G. Maas, Validation of geometric models for fisheye lenses, ISPRS J. Photogramm. Remote Sens. 64 (2009) 259–266 (Theme Issue: Image Analysis and Image Engineering in Close Range Photogrammetry).

[3] X. Zhang, J. Stockel, M. Wolf, P. Cathier, G. McLennan, E.A. Hoffman, M. Sonka, A new method for spherical object detection and its application to computer aided detection of pulmonary nodules in CT images, in: Medical Image Computing and Computer-Assisted Intervention (MICCAI) at International Conference on Medical Image Computing and Computer-Assisted Intervention, vol. 10, 2007, pp. 842–849.

[4] H. Skibbe, M. Reisert, H. Burkhardt, SHOG – spherical HOG descriptors for rotation invariant 3d object detection, in: R. Mester, M. Felsberg (Eds.), Pattern Recognition, Lecture Notes in Computer Science, vol. 6835, Springer, Berlin Heidelberg, 2011, pp. 142–151.

[5] R. Mittal, P. Bhowmick, SPReAD: on spherical part recognition by axial discretization in 4d hough space, in: L. Bolc, R. Tadeusiewicz, L. Chmielewski, K. Wojciechowski (Eds.), Computer Vision and Graphics, Lecture Notes in Computer Science, vol. 7594, Springer, Berlin Heidelberg, 2012, pp. 188–195.

[6] M. Kharbat, N. Aouf, A. Tsourdos, B. White, Sphere detection and tracking for a space capturing operation, in: IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS), 2007, pp. 182–187.

[7] A. Fazlollahi, F. Meriaudeau, V.L. Villemagne, C.C. Rowe, P.M. Desmond, P.A. Yates, O. Salvado, P. Bourgeat, Automatic detection of small spherical lesions using multiscale approach in 3D medical images, in: 2013 IEEE International Conference on Image Processing, 2013, pp. 1158–1162.

[8] H. Skibbe, M. Reisert, O. Ronneberger, H. Burkhardt, Spherical bessel filter for 3D object detection, in: 2011 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2011, pp. 728–733.

[9] A. Abuzaina, M. Nixon, J. Carter, Sphere detection in kinect point clouds via the 3d hough transform, in: R. Wilson, E. Hancock, A. Bors, W. Smith (Eds.), Computer Analysis of Images and Patterns, Lecture Notes in Computer Science, vol. 8048, Springer, Berlin Heidelberg, 2013, pp. 290–297.

[10] A.J.P. Daniel, A. Martins, Antnio J.R. Neves, Real-time generic ball detection in RoboCup domain, in: IBERAMIA'08 – Ibero-American Conference on Artificial Intelligence: IROBOT'08 – 3rd International Workshop on Intelligent Robotics, Lisboa, Portugal, 2008, pp. 37–48.

[11] A. Voigtlnder, S. Lange, M. Lauer, M.A. Riedmiller, Real-time 3d ball recognition using perspective and catadioptric cameras, in: European Conference on Mobile Robotics (EMCR), Freiburg, Germany, 2007.

[12] A. Neves, J.L. Azevedo, B. Cunha, J. Cunha, R. Dias, P. Fonseca, N. Lau, E. Pedrosa, A. Pereira, J. Silva, Cambada'12: Team description paper, in: RoboCup 2012, Mexico City, Mexico, 2012.

[13] M. Wenig, K. Pang, P. On, Arbitrarily colored ball detection using the structure tensor technique, Mechatronics 21 (2011) 367–372 (Special Issue on Advances in intelligent robot design for the Robocup Middle Size League).

[14] M. Taiana, J. Santos, J. Gaspar, J. Nascimento, A. Bernardino, P. Lima, Tracking objects with generic calibrated sensors: an algorithm based on color and 3d shape features, Robot. Autonom. Syst. 58 (2010) 784–795 (Omnidirectional Robot Vision).

[15] A. Ahmad, P. Lima, Multi-robot cooperative spherical-object tracking in 3d space based on particle filters, Robot. Autonom. Syst. 61 (2013) 1084–1093.

[16] A. Ahmad, G.D. Tipaldi, P. Lima, W. Burgard, Cooperative robot localization and target tracking based on least square minimization, in: Proceedings of the 2013 IEEE International Conference on Robotics and Automation, 2013. ICRA, 2013.

[17] A. Ahmad, T. Nascimento, A.G.S. Conceição, A.P. Moreira, P. Lima, Perception-driven multi-robot formation control, in: Proceedings of the 2013 IEEE International Conference on Robotics and Automation, 2013. ICRA, 2013.

[18] S. Thrun, W. Burgard, D. Fox, Probabilistic Robotics (Intelligent Robotics and Autonomous Agents), The MIT Press, 2005.

[19] H. Silva, A. Dias, J. Almeida, A. Martins, E. Silva, Real-time 3D ball trajectory estimation for RoboCup middle size league using a single camera, Lecture Notes in Computer Science, vol. 7416, Springer, Berlin/Heidelberg, 2012.