Moving object detection, tracking and following using an omnidirectional camera on a mobile robot

Ivan Markovic, François Chaumette, Ivan Petrovic

Moving object detection, tracking and following using an omnidirectional camera on a mobile robot*

Ivan Marković[1], François Chaumette[2] and Ivan Petrović[1]

***Abstract*— Equipping mobile robots with an omnidirectional camera is very advantageous in numerous applications as all information about the surrounding scene is stored in a single image frame. In the given context, the present paper is concerned with detection, tracking and following of a moving object with an omnidirectional camera. The camera calibration and image formation is based on the spherical unified projection model thus yielding a representation of the omnidirectional image on the unit sphere. Detection of moving objects is performed by calculating a sparse optical flow in the image and then lifting the flow vectors on the unit sphere where they are discriminated as dynamic or static by analytically calculating the distance of the terminal vector point to a great circle arc. The flow vectors are then clustered and the center of gravity is calculated to form the sensor measurement. Furthermore, the tracking is posed as a Bayesian estimation problem on the unit sphere and the solution based on the von Mises-Fisher distribution is utilized. Visual servoing is performed for the object following task where the control law calculation is based on the projection of a point on the unit sphere. Experimental results obtained by a camera with a fish-eye lens mounted on a differential drive mobile robot are presented and discussed.**

I. INTRODUCTION

Omnidirectional cameras by their definition provide a 360° view of the surrounding scene and as such pose themselves as a powerful tool in robot's vision system. The enhanced field of view can be obtained by using several synchronized panoramic cameras, a combination of a camera and a mirror, or a camera with a wide-angle lens. The amount of information in such a single image reinforces robot's abilities in interpreting and adequately acting and reacting in the environment. The sensor has been utilized in mobile robotics in a variety of applications: visual odometry, navigation, structure-from-motion, visual servoing, and moving object tracking to name but a few.

Detection and tracking of moving objects with a camera mounted on a mobile robot is a task inconvenienced by the simultaneous ego-motion of the robot and the motion of the objects. With perspective cameras the problem is approached in [1] by calculating the optical flow and optimizing the bilinear transformation to warp the image between the consecutive frames, after which the images are differentiated and motion is detected. Then the particle filter is used to track the moving objects in the image and a laser range finder is used to infer about the location in 3D. In [2] the detection was based on monocular scene reconstruction and affine transformation of a triangle mesh in order to perform the image warping. The tracking of the moving object and the scene reconstruction was performed using the extended Kalman filter. Compared to perspective cameras, omnidirectional cameras offers the advantage of being able to address objects in the 360° surrounding scene of the mobile robot, thus, for example, removing the possibility that the followed object will escape the camera's field-of-view.

In [3] and [4] an omnidirectional image was first unwrapped to a panoramic image using a cylindrical projection, where the optical flow is calculated. In the former a synthetic optical flow is generated by estimating the position of the focii of expansion and contraction and the calculated flow is compared to the generated one, while the latter estimates an affine transform on square subsegments to warp the image and perform the differentiation. In [5] the omnidirectional image is segmented in a set of perspective images and detection is done in the vein of [1], while the tracking is based on the particle filter. To perform the following a control law based on a minimization of an ad hoc following error is calculated. Compared to cylindrical projection geometry, working on the unit sphere offers the advantage of mitigating wrapping issues at the edges of the projection caused by cutting and unwrapping the cylinder.

In the present paper, excepting the optical flow calculation, we propose a method for moving object detection, tracking and following based on processing on the unit sphere thus taking into account the specific sensor geometry and making it as general as possible for calibrated omnidirectional systems, since any such image can be represented on the unit sphere. The novel moving object detection is based on analytically calculating the distance of a point on the unit sphere to a great circle arc on the unit sphere. The tracking is performed in a Bayesian prediction-correction manner where the underlying distribution is a distribution on the unit sphere, namely the von Mises-Fisher distribution [6]. In the end, the object following is based on visual servoing [7] where the control law is calculated from an interaction matrix derived for a projection of a point on the unit sphere. In our previous work [8] a mixture of von Mises distributions (distribution on the unit circle) was proposed as a Bayesian method for speaker azimuth estimation with a microphone array.

*This work has been supported by European Community's Seventh Framework Programme under grant agreement no. 285939 (ACROSS). The research related to object following by visual servoing was supported by the University of Zagreb as a short term financial support under the contract number 2013-ZUID-19.

[1]Ivan Marković and Ivan Petrović are with University of Zagreb, Faculty of Electrical Engineering and Computing, Department of Control and Computer Engineering, 10000 Zagreb, Croatia (e-mail: ivan.markovic, ivan.petrovic@fer.hr)

[2]François Chaumette is with Inria Rennes-Bretagne Atlantique, Campus de Beaulieu, Rennes, Cedex 35042, France (e-mail: francois.chaumette@irisa.fr)

II. CAMERA CALIBRATION

The unified projection model describes the image formation in catadioptric systems with a unique effective viewpoint, which includes combinations of the mirror—parabolic, hyperbolic, elliptic, planar—and the lens—orthographic or perspective. Theoretical derivation of complete single-lens single-mirror catadioptric sensors characterized by a unique effective viewpoint was introduced in [9], while the unified projection model was introduced and studied in [10], [11]. In the present paper we have chosen to use the calibration method based on planar grids proposed in [12]. Although the model and the calibration methods were developed for systems with a unique effective viewpoint, in practice they have been shown to be valid for dioptric systems with a fish-eye lens [13].

Consider a point in space $\mathbf{P}$ and a frame $\mathcal{F}_o : (x_o, y_o, z_o)$ as shown in Fig. 2. First, $\mathbf{P}$ is projected to the surface of the sphere. Then, the normalized point $\mathbf{P}_n$ is perspectively projected from the coordinate system $\mathcal{F}_m : (x_m, y_m, z_m)$ to the point $\mathbf{m} = (x, y, 1)$ on the normalized plane. The point in the image $\mathbf{p}$ is obtained by calculating $\mathbf{p} = \mathbf{K}\mathbf{m}$, where $\mathbf{K}$ is a 3×3 matrix containing the camera intrinsic parameters. The matrix $\mathbf{K}$ and ξ are obtained by the calibration procedure. In the present paper we used a standard perspective camera with a fish-eye lens, and the best calibration results were obtained by utilizing just the $\mathbf{K}$ and ξ, i.e. without distortion modeling. In the sequel we assume that our omnidirectional camera is calibrated, which consequently enables us to apply the inverse projection (lifting) of the point in the image to a point on the unit sphere [12]

$$\mathbf{m} = \mathbf{K}^{-1}\mathbf{p}, \quad \mathbf{P}_n = \begin{bmatrix} \frac{\xi+\sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1}x \\ \frac{\xi+\sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1}y \\ \frac{\xi+\sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1} - \xi \end{bmatrix}. \quad (1)$$

III. MOVING OBJECT DETECTION

The main goal of our vision system is to detect moving objects in the omnidirectional image while the robot itself moves. However, this proves to be a daunting task since we have motion in the image induced both by the moving objects and the ego-motion of the robot. We have approached this problem in one of our previous works [14] by estimating the sparse optical flow in the image and by using the robot's odometry to discriminate between the flow vectors induced by the ego-motion (static features) from those induced by the moving objects (dynamic features). The detection part in the present paper continues in the similar vein, but with several important distinctions—after the optical flow is calculated in the image, the higher-level processing is done on the sphere and vector discrimination is performed analytically as opposed to by iteratively projecting points from different heights.

More concretely, we calculate the optical flow directly in the camera image coordinates using the sparse iterative version of the Lucas-Kanade algorithm in pyramids [15] implemented in the OpenCV library [16]. Furthermore, both

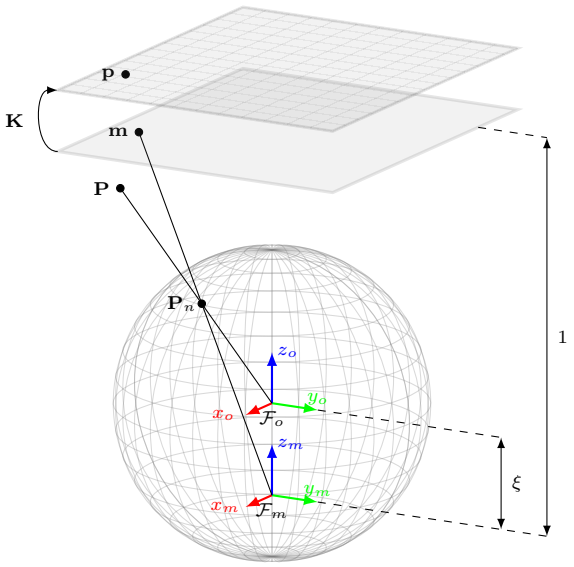

Fig. 2. Illustration of the unified image formation

the initial point (feature position in the previous frame) and the terminal point (feature position in the current frame) of the optical flow vector are lifted to the unit sphere for further processing. An extension and improvement would be to calculate the optical flow and the low-level image processing on the unit sphere/Riemannian manifolds [17] since it has been shown that operators based on the spherical image yield better results than operators derived for perspective images [18], [19], [17].

With the optical flow calculated, we need to devise a procedure for finding the optical flow vectors caused by the moving objects. Consider Fig. 1 where we have depicted a sphere with $\mathcal{F}_p : (x_p, y_p, z_p)$ coordinate system in the origin—representing the image in the previous frame, henceforth referred to as the previous sphere—and a second sphere with $\mathcal{F}_c : (x_c, y_c, z_c)$ coordinate system in the origin—representing the image in the current frame, henceforth referred to as the current sphere. We assume that the displacement between the previous and the current sphere, i.e. $\mathcal{F}_p$ and $\mathcal{F}_c$, is known (in practice calculated from odometry measurements) and described by ${}^c\mathbf{R}_p$ and ${}^c\mathbf{t}_p$ accounting for the rotation and the translation, respectively. Furthermore, the point ${}^p\mathbf{P}$ in $\mathcal{F}_p$ represents a lifted point detected in the previous image whose matched point in the current image, the lifted point ${}^c\mathbf{P}_m$ in $\mathcal{F}_c$, has been determined by the optical flow algorithm. To determine whether this optical flow vector was induced by the moving object or the ego-motion, we will first hypothesize that the flow was due to ego-motion, and then if the condition is not met we will classify it as being caused by the moving object.

In order to achieve this task, we need to know where to expect a static feature from the previous sphere, like ${}^p\mathbf{P}$, on the current sphere (of course without any information about the depth of the feature). By looking at Fig. 1 we can assert that the point ${}^p\mathbf{P}$ is projection of a feature in the environment somewhere along the optical ray defined

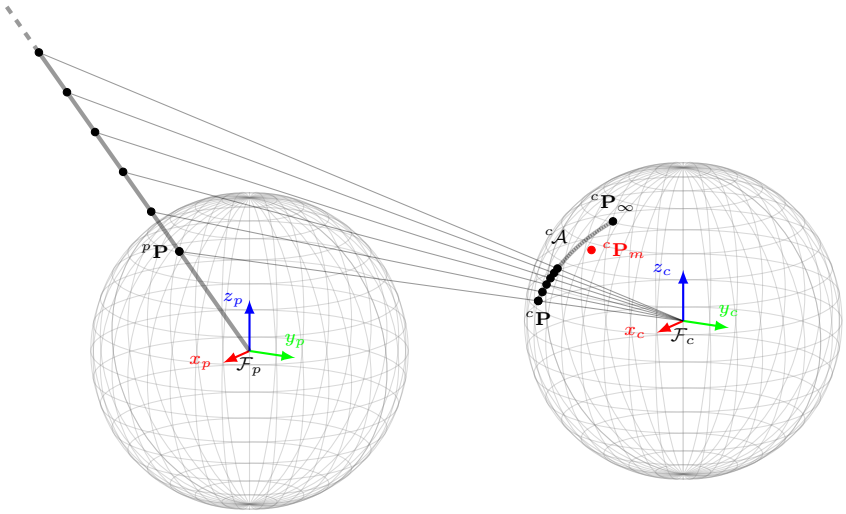

Fig. 1. Hypothetical location on the sphere in the current frame of the feature on the sphere in the previous frame

by the previous sphere's origin of $\mathcal{F}_p$ and the point ${}^{p}\mathbf{P}$. Furthermore, projection of the point ${}^{p}\mathbf{P}$ onto the current sphere is the point ${}^{c}\mathbf{P}$, and if we continued along the ray in $\mathcal{F}_p$ we can see in Fig. 1 where the points would project to on the current sphere. The point on the ray in the infinity projects to ${}^{c}\mathbf{P}_\infty = {}^{c}\mathbf{R}_p {}^{p}\mathbf{P}$, i.e. as if the point ${}^{p}\mathbf{P}$ did not move at all except for the rotation. Given the previous analysis we can conclude that a projection of a point like ${}^{p}\mathbf{P}$, representing a static feature on the previous sphere, should theoretically lie somewhere along the arc ${}^{c}\mathcal{A}$ of the great circle ${}^{c}\mathcal{C}$ [1] defined by points ${}^{c}\mathbf{P}$ and ${}^{c}\mathbf{P}_\infty$. To conclude, we will classify an optical flow vector as induced by ego-motion if its matched point on the current sphere ${}^{c}\mathbf{P}_m$ lies close to the aforementioned great circle arc ${}^{c}\mathcal{A}$. Naturally, this approach cannot detect objects moving along the optical ray, but this is an unlikely event since it is not possible to ensure such a scenario for all points belonging to a rigid object.

In spherical geometry the closest distance between two points on the sphere is the so called great circle distance and for unit spheres it can be directly calculated as

$$d({}^{c}\mathbf{P},{}^{c}\mathbf{P}_\infty) = \arccos\left({}^{c}\mathbf{P} \cdot {}^{c}\mathbf{P}_\infty\right), \tag{2}$$

where $(\cdot)$ represents the scalar product. Equation (2) is simply the angle between the two unit vectors and incidentally the length of the arc ${}^{c}\mathcal{A}$ in Fig. 1. In order to calculate the distance of ${}^{c}\mathbf{P}_m$ to ${}^{c}\mathcal{A}$, we first need to determine a point ${}^{c}\mathbf{Q}_m$ on the great circle ${}^{c}\mathcal{C}$ which is closest to ${}^{c}\mathbf{P}_m$ [20]. We solve this by projecting ${}^{c}\mathbf{P}_m$ to the plane defined by ${}^{c}\mathbf{P}$ and ${}^{c}\mathbf{P}_\infty$ and then normalizing it to obtain a unit vector

$$\mathbf{P}' = {}^{c}\mathbf{P}_m - \left({}^{c}\mathbf{P}_m \cdot \mathbf{n}\right)\mathbf{n}, \quad {}^{c}\mathbf{Q}_m = \frac{\mathbf{P}'}{|\mathbf{P}'|}, \tag{3}$$

where $\mathbf{n} = {}^{c}\mathbf{P} \times {}^{c}\mathbf{P}_\infty$ and $(\times)$ represents the vector product. At this stage we have two possible positions of the point ${}^{c}\mathbf{Q}_m$: it either lies on ${}^{c}\mathcal{A}$, or outside of it but on ${}^{c}\mathcal{C}$. The former case is true if the point ${}^{c}\mathbf{P}_m$ lies in the lune[2] of ${}^{c}\mathcal{A}$ which we verify by testing the following condition [20]

$$\begin{aligned} &({}^{c}\mathbf{P} \times {}^{c}\mathbf{Q}_m) \cdot ({}^{c}\mathbf{Q}_m \times {}^{c}\mathbf{P}_\infty) > 0 \quad \text{and} \\ &({}^{c}\mathbf{P} \times {}^{c}\mathbf{Q}_m) \cdot ({}^{c}\mathbf{P} \times {}^{c}\mathbf{P}_\infty) > 0. \end{aligned} \tag{4}$$

Thus if ${}^{c}\mathbf{Q}_m$ lies on ${}^{c}\mathcal{A}$ the distance of the point ${}^{c}\mathbf{P}_m$ to the arc ${}^{c}\mathcal{A}$ is calculated as $d({}^{c}\mathbf{P}_m,{}^{c}\mathbf{Q}_m)$, otherwise as $\min\{d({}^{c}\mathbf{P}_m,{}^{c}\mathbf{P}), d({}^{c}\mathbf{P}_m,{}^{c}\mathbf{P}_\infty)\}$. If the robot does not move or just rotates then condition (4) is evaluated as false and $d({}^{c}\mathbf{P},{}^{c}\mathbf{P}_m)$ is calculated.

The detection performance depends on the measured displacement between two consecutive images. In the present paper we have utilized wheels' odometry for the task, but this can be further refined by fusion with other sensors, like the inertial measurement unit, or by combining the odometry with image-based methods like omnidirectional visual odometry and simultaneous localization and mapping (SLAM).

Once we have selected optical flow vectors that we consider to be caused by moving objects, we still need to make sense of that particular set. The vectors are partitioned in equivalence classes using disjoint set data structure and union find algorithm. We state that two vectors belong to the same group if they have similar modulo, elevation and azimuth (note that the vectors are compared after being lifted to the sphere). The formed groups of vectors are treated as representing moving objects in the scene and their center of gravity is calculated. The center of the group is then a vector on the unit sphere which we henceforth treat as our sensor measurement.

IV. TRACKING ON THE UNIT SPHERE

At this stage we are working with vectors on the unit sphere which represent the direction of the detected moving objects. We propose at this point to advance by statistically

[1] Intersection of the sphere and a plane which passes through the center point of the sphere

[2] Area on a sphere bounded by two half great circles.

modeling the measured direction, i.e. to pose a probabilistic model of the sensor measurement, by using a directional statistics distribution on the unit sphere, namely the von Mises-Fisher (vMF) distribution.

In general the vMF is a distribution on a $(p-1)$-dimensional sphere in $\mathbb{R}^p$. In our case $p = 3$ and the distribution has the following form [21]

$$p(\mathbf{x};\kappa,\boldsymbol{\mu}) = \frac{\kappa}{4\pi \sinh \kappa} \exp\left(\kappa \boldsymbol{\mu}^{\mathrm{T}}\mathbf{x}\right), \tag{5}$$

where $\boldsymbol{\mu}$ is the mean direction and κ is the concentration parameter. Because (5) is symmetrical about $\boldsymbol{\mu}$, the mean direction of $\mathbf{x}$ is $\boldsymbol{\mu}$. For $\kappa > 0$, the distribution has a mode at the mean direction μ, whereas when $\kappa = 0$ the distribution is uniform. The larger the κ the greater the clustering around the mean direction. Since (5) depends on $\mathbf{x}$ solely through $\boldsymbol{\mu}^{\mathrm{T}}\mathbf{x}$, the vMF is rotationally symmetric about $\boldsymbol{\mu}$. Historically, the vMF first appeared in statistical mechanics in the context of weakly interacting dipoles subjected to an external electric field, whose directions tend to have a vMF distribution [21].

Furthermore, we pose the problem as an estimation on a sphere thus devising a Bayesian state estimator (tracker) based on the vMF distribution [6]. This consists of constructing the *a posteriori* probability density function (pdf) of the state based on all available information. A Bayesian estimation procedure of the a posteriori pdf consists of two steps: prediction and update [22], which in our case entails representing the state to be estimated $\mathbf{x}_k$ at time k as the vMF distribution and successively predicting and updating this distribution. The prediction step involves calculating the pdf via the total probability theorem

$$p(\mathbf{x}_k|\mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})\, \mathrm{d}\mathbf{x}_{k-1}, \tag{6}$$

where $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is the probabilistic model of the state evolution, $p(\mathbf{x}_{k-1}|\mathbf{z}_{1:k-1})$ is the posterior at time $k-1$, and $\mathbf{z}_{1:k-1}$ are all measurements up to and including time $k-1$. In our case we choose to add process noise governed by a centered vMF in the prediction stage which amounts to convolving our posterior at time $k-1$ with the vMF distribution representing the process noise. Given two vMF distributions, $p(\mathbf{x};\kappa_i,\boldsymbol{\mu})$ and $p(\mathbf{x};\kappa_j,\boldsymbol{\mu})$, the result of the convolution does not produce another vMF distribution. However, the result of this operation can be well approximated by a vMF with unchanged mean direction and a suitably chosen value of the resulting κ [21]

$$\kappa_{ij} = A^{-1}(A(\kappa_i)A(\kappa_j)), \quad A(\kappa) = \frac{1}{\tanh \kappa} - \frac{1}{\kappa}. \tag{7}$$

In the update step, the posterior at time k is calculated via the Bayes theorem

$$p(\mathbf{x}_k|\mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{1:k-1})}, \tag{8}$$

where $p(\mathbf{z}_k|\mathbf{x}_k)$ is the sensor model and $p(\mathbf{z}_k|\mathbf{z}_{1:k-1})$ is the normalizer. In our case the sensor model $p(\mathbf{z}_k|\mathbf{x}_k)$ will be represented by a vMF where the measurement $\mathbf{z}_k$ is the center of gravity of the dynamic vector cluster, while the predicted state $p(\mathbf{x}_k|\mathbf{z}_{1:k-1})$ will be the result of the previously discussed convolution. Given two vMF distributions, $p(\mathbf{x};\kappa_i,\boldsymbol{\mu}_i)$ and $p(\mathbf{x};\kappa_j,\boldsymbol{\mu}_j)$, the result of the update step is a vMF with the following parameters [6]

$$\kappa_{ij} = \sqrt{\kappa_i^2 + \kappa_j^2 + 2\kappa_i\kappa_j(\boldsymbol{\mu}_i \cdot \boldsymbol{\mu}_j)}, \quad \boldsymbol{\mu}_{ij} = \frac{\kappa_i\boldsymbol{\mu}_i + \kappa_j\boldsymbol{\mu}_j}{\kappa_{ij}}. \tag{9}$$

These two steps, governed by (7) and (9), will cyclically produce the estimate of the direction of the moving object. Methods for practical calculation of some of the aforementioned equations involving the vMF distribution can be found in [23].

In this paper we focus on tracking a single object and if there are multiple moving objects detected, then only the closest measurement is considered in the update step. Fig. 3 depicts the measured and estimated direction azimuth and elevation of the moving object from an experiment in which an object circled around the mobile robot. From the figure we can see that the elevation measurements are more noisy than the azimuth measurements, and that the filter manages to smoothly track the moving object. This is important since we need smooth estimates for the control task.

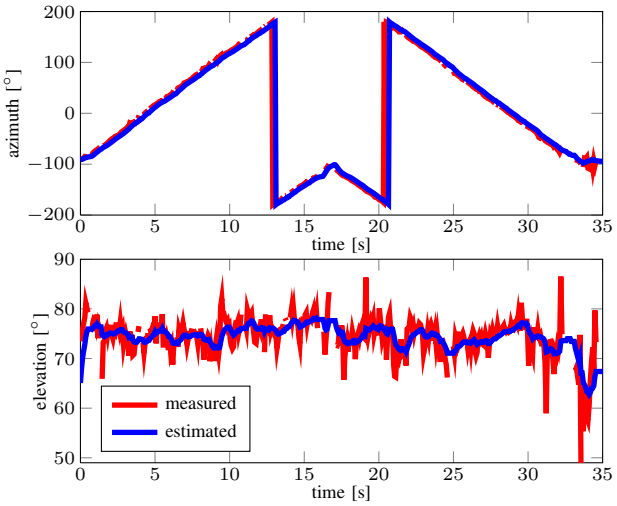

Fig. 3. Azimuth and elevation of the moving object direction

V. OBJECT FOLLOWING BY VISUAL SERVOING

The idea of the following task is to keep the tracked moving object at the specific (user-defined) location in the omnidirectional image. In the present paper we have utilized a differential drive mobile robot where the linear and the angular velocity are controlled, hence our task at hand is to calculate the control law that will drive the error between the desired and the estimated direction to zero. To solve this problem we propose to utilize a visual servoing technique based on the projection of a point on the unit sphere.

We used a cylindrical coordinate system in the spherical image to represent the visual feature $\mathbf{s} = (s_x, s_y)$, i.e. the image-plane projection of the estimated direction $\mathbf{x}_k$ on the

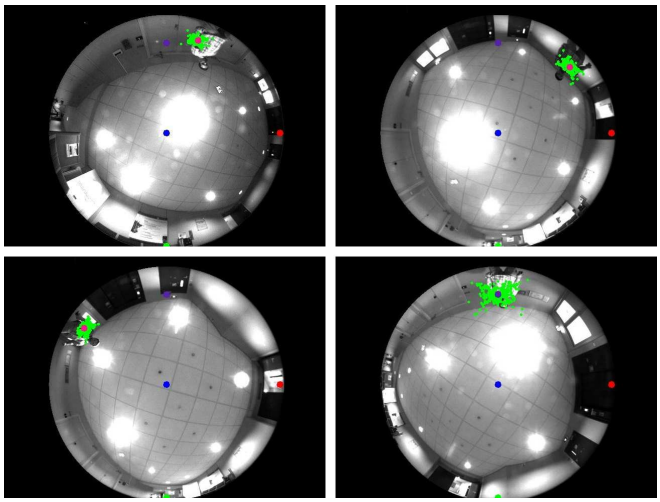

Fig. 4. Snapshots of an experiment. Upper left image is the earliest in time, while the lower right is the latest in time.

unit sphere

$$\rho = \sqrt{s_x^2 + s_y^2}, \quad \theta = \arctan \frac{s_y}{s_x}. \tag{10}$$

For differential drive robots the convention is to set the robot's coordinate system such that the linear velocity v is in the positive direction of the x axis, while the angular velocity ω is defined positive counter-clockwise with respect to the z axis. Let the spatial velocity of the camera be denoted by $\boldsymbol{v} = (v, \omega)$. The relationship between $\dot{\boldsymbol{s}}$ and $\boldsymbol{v}$ is given by $\dot{\boldsymbol{s}} = \mathbf{L}_s \boldsymbol{v}$, where the interaction matrix evaluates to [24]

$$\mathbf{L}_s = \begin{bmatrix} \frac{-\cos\theta}{P_z} & 0 \\ \frac{\sin\theta}{\rho P_z} & -1 \end{bmatrix}, \tag{11}$$

and P_z is the z coordinate of the moving object (not on the sphere but in the environment). Then, the calculation of the control law proceeds as follows [7]

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = -\lambda \hat{\mathbf{L}}_s^{-1} \begin{bmatrix} \rho - \rho^* \\ \theta - \theta^* \end{bmatrix}, \tag{12}$$

where λ is a positive gain, ρ^* and θ^* are the desired values and $\hat{\mathbf{L}}_s^{-1}$ is the inverse of the estimated interaction matrix—the coordinate P_z is not known and in the present paper we set it to an arbitrary value. The gain λ is adaptively calculated according to the following law

$$\lambda(e) = a \exp(-be) + c, \tag{13}$$

where e is the error of the control task (calculated as the great circle distance between the estimated and the desired position), $a = \lambda(0) - \lambda(\infty), b = \lambda'(0)/a, c = \lambda(\infty)$ with $\lambda(0) = 0.5, \lambda(\infty) = 0.05, \lambda'(0) = 0.5$.

In conclusion, the estimated direction $\mathbf{x}_k$ is projected onto the image plane and represented in cylindrical coordinates (ρ, θ) and the control is then calculated via (12). Visual servoing control law in spherical coordinates [25], for our specific task with the differential drive robot, exhibits an additional singularity in the control law compared to the representation in cylindrical coordinates. Namely, in the cylindrical coordinate system as in [24] the control law has a singularity in $\rho = 0$—estimated direction in the middle of the image (azimuth undefined, in our case practically unlikely)—and $\theta = \pm\pi/2$—the values of the azimuth angle (possible in our case if the estimated direction gets too far away from the desired one; in that case we saturate the control signal to a reasonable maximal value). However, the control law in spherical coordinates [25] has additional singularity when the elevation of the estimated direction is equal to $\pm\pi/2$—the vector lies on the sphere's equator (fairly often in our case). This was the reasoning due to which we chose to work in the cylindrical coordinate system.

VI. EXPERIMENTAL RESULTS

The experiments were carried out on a Pioneer 3DX differential drive mobile robot equipped with an omnidirectional camera composed out of a Point Grey Dragonfly2 camera and an Omnitech Robotics fish-eye lens. The task of the robot was to detect a moving object in the image, track it and use the visual servoing control law to follow the object

A snapshot of an experiment is shown in Fig. 4. Camera's coordinate system rotated for $\pi/2$ from the robot's system is depicted by red, green and blue points (projections of the tips of the coordinate axes in the sphere). The violet point is the desired direction of the moving object, while the magenta point represents the estimated direction of the moving object. The green points surrounding the estimated direction are samples from the posterior vMF representing the current state of the object.

In Fig. 5, for three experiments, we have depicted linear and angular velocity commands together with the visual servoing task error, which is calculated as the great circle distance between the desired and the estimated direction of the object via (2). In the experiments the object moved so as to first distance itself from the desired position and then waited until the robot closed the distance by reducing the servoing task error to zero (see the accompanying video). This motion pattern was repeated several times during the experiments and the result can be clearly seen in the error depicted in Fig. 5. Concerning the control velocities we can see that the angular velocity follows closely the behavior of the error—when the error is greatest so is the velocity command, sometimes changing the sign of the command (when the robot would correct for the error the object would move away in the direction from which the robot came) and sometimes keeping the same sign (when the robot would correct the error coming from one direction the object would move away in the opposite). The linear velocity command is more noisy than its angular counterpart due to the fact that our visual servo control law corrects the error based on the projection of a single point. We have no information about the shape nor the height of the object which makes it difficult to guarantee that the robot would position itself relative to the object at a specific distance. However, in practice we have noticed that most often the tracked center of gravity would not deviate much thus making it possible to define a point in the image for which the robot would reasonably close the

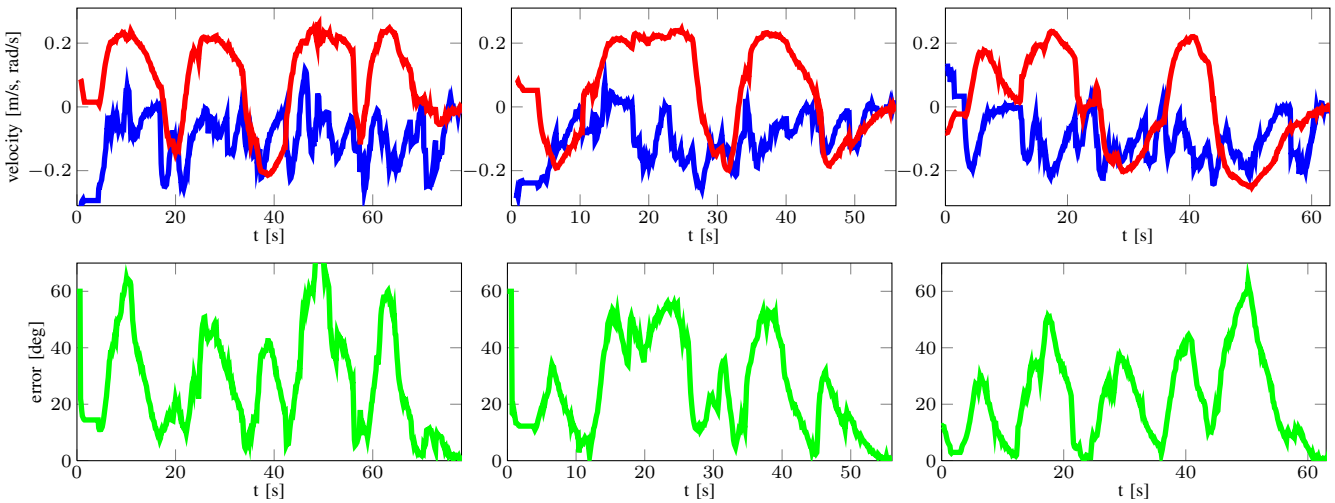

Fig. 5. Command velocities—linear (red) and angular (blue), and error of the control task (great circle distance from the desired to the estimated direction)

distance to the object. Naturally, none of the aforementioned problems were experienced with the orientation.

VII. CONCLUSION

We have presented a method based on processing on the unit sphere for moving object detection, tracking and following with an omnidirectional camera mounted on a mobile robot. The spherical projection model coupled with odometry-based displacement information was used to segment out vectors that do not belong to the static scene around the mobile robot. The center of gravity of the dynamic clusters was then probabilistically structured and included in the tracking framework based on the Bayesian estimation on the sphere with the von Mises-Fisher distribution. Given the estimated position a control law based on visual servoing was calculated which in turn made the robot follow the moving object. Experimental results obtained with a camera and fish-eye lens mounted on a differential drive platform were presented and discussed.

REFERENCES

[1] B. Jung and G. Sukhatme, "Real-time motion tracking from a mobile robot," *International Journal of Social Robotics*, vol. 2, no. 1, pp. 63–78, 2009.

[2] E. Einhorn, M. Filzhuth, C. Schröter, and H.-M. Gross, "Monocular detection and estimation of moving obstacles for robot navigation," in *European Conference on Mobile Robots (ECMR)*, 2011, pp. 121–126.

[3] J. Kim and Y. Suga, "An omnidirectional vision-based moving obstacle detection in mobile robot," *International Journal of Control, Automation, and Systems*, vol. 5, no. 6, pp. 663–673, 2007.

[4] C.-M. Oh, Y.-C. Lee, D.-Y. Kim, and C.-W. Lee, "Moving object detection in omnidirectional vision-based mobile robot," *Annual Conference on IEEE Industrial Electronics Society (IECON)*, pp. 4232–4235, 2012.

[5] M. Kobilarov, G. Sukhatme, J. Hyams, and P. Batavia, "People tracking and following with mobile robot using an omnidirectional camera and a laser," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2006, pp. 557–562.

[6] A. Chiuso and G. Picci, "Visual tracking of points as estimation on the unit sphere," *The Confluence of Vision and Control, Lecture Notes in Control and Information Sciences*, vol. 237, pp. 90–105, 1998.

[7] F. Chaumette and S. Hutchinson, "Visual Servoing and Visual Tracking," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2008, ch. Visual Servoing, pp. 563–583.

[8] I. Marković and I. Petrović, "Bearing-only tracking with a mixture of von Mises distributions," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 707–712.

[9] S. Baker and S. Nayar, "A theory of catadioptric image formation," in *International Conference on Computer Vision (ICCV)*, 1998, pp. 35–42.

[10] C. Geyer and K. Daniilidis, "A unifying theory for central panoramic systems and practical implications," in *European Conference on Computer Vision (ECCV)*, 2000, pp. 445–461.

[11] P. J. Barreto and H. Araújo, "Issues on the geometry of central catadioptric image formation," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001, pp. 422–427.

[12] C. Mei and P. Rives, "Single view point omnidirectional camera calibration from planar grids," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 3945–3950.

[13] X. Ying and Z. Hu, "Can we consider central catadioptric cameras and fisheye cameras within a unified imaging model?" in *European Conference on Computer Vision (ECCV)*, 2004, pp. 442–455.

[14] D. Herceg, I. Marković, and I. Petrović, "Real-time detection of moving objects by a mobile robot with an omnidirectional camera," in *International Symposium on Image and Signal Processing and Analysis (ISPA)*, 2011, pp. 289–294.

[15] J.-Y. Bouguet, "Pyramidal implementation of the Lucas Kanade feature tracker description of the algorithm," Intel Corporation, Microsoft Research Labs, Tech. Rep. 2, 2000.

[16] G. Bradski, "The OpenCV library," *Dr. Dobb's Journal of Software Tools*, 2000.

[17] L. Puig and J. J. Guerrero, "Scale space for central catadioptric systems: Towards a generic camera feature extractor," in *IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 1599–1606.

[18] C. Demonceaux and P. Vasseur, "Omnidirectional image processing using geodesic metric," in *IEEE International Conference on Image Processing (ICIP)*, 2009, pp. 221–224.

[19] H. Hadj-Abdelkader, E. Malis, and P. Rives, "Spherical image processing for accurate visual odometry with omnidirectional cameras," in *The 8th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras (OMNIVIS)*, 2008.

[20] A. Brink, "Speeding up the computation of similarity measures based on Minkowski addition in 3D," p. 40, 2004.

[21] K. V. Mardia and P. E. Jupp, *Directional Statistics*. New York: Wiley, 1999.

[22] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, 2006.

[23] W. Jakob, "Numerically stable sampling of the von Mises-Fisher distribution on S^2 (and other tricks)," Interactive Geometry Lab, ETH Zürich, Tech. Rep., 2012.

[24] R. Fomena and F. Chaumette, "Improvements on visual servoing from spherical targets using a spherical projection model," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 874–886, Aug. 2009.

[25] P. Corke, "Spherical image-based visual servo and structure estimation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 5550–5555.