Gaze Tracing in a Bounded Log-spherical Space for Artificial Attention Systems

Beatriz Oliveira[1], Pablo Lanillos[1], and João Filipe Ferreira[1]

AP4ISR team, Institute of Systems and Robotics (ISR)
University of Coimbra. Pinhal de Marrocos, Polo II, 3030-290 COIMBRA, Portugal
{beatriz, planillos, jfilipe}@isr.uc.pt
http://ap.isr.uc.pt http://mrl.isr.uc.pt/projects/casir

Abstract. Human gaze is one of the most important cue for social robotics due to its embedded intention information. Discovering the location or the object that an interlocutor is staring at, gives the machine some insight to perform the correct attentional behaviour. This work presents a fast voxel traversal algorithm for estimating the potential locations that a human is gazing. Given a 3D occupancy map in log-spherical coordinates and the gaze vector, we evaluate the regions that are relevant for attention by computing the set of intersected voxels between an arbitrary gaze ray in the 3D space and a log-spherical bounded section defined by $\rho \in (\rho_{min}, \rho_{max}), \theta \in (\theta_{min}, \theta_{max}), \phi \in (\phi_{min}, \phi_{max})$. The first intersected voxel is computed in closed form and the rest are obtained by binary search guaranteeing no repetitions in the intersected set. The proposed method is motivated and validated within a human-robot interaction application: gaze tracing for artificial attention systems.

Keywords: human-robot interaction (HRI), artificial attention, gaze tracing, log-spherical, voxel traversal algorithm

1 Introduction

With the forthcoming social or assisted living robotic platforms, interlocutor gaze has become a valuable cue to interpret intentionality [3, 11]. The next generation of robotic platforms should be able to interpret and generate social signals [17] by means of non-verbal communication. This emotional, body and attentional language will help roboticists to develop low cost robots with high user acceptance. In this sense, the direction of other's attention is crucial for mastering social interaction [16]. In fact, in the case of joint attention [3, 7], where two individuals perform a triadic relationship between them and an object, gaze direction provides useful information to know if the human is looking at the robot as well as about the direction of the shared object [2]. This location is potentially important for the correct social interaction and can also play a significant role in the task being effectuated.

Figure 1 describes a generic and simple example of social interplay within human-robot interaction (HRI) in assisted living applications. An old lady is

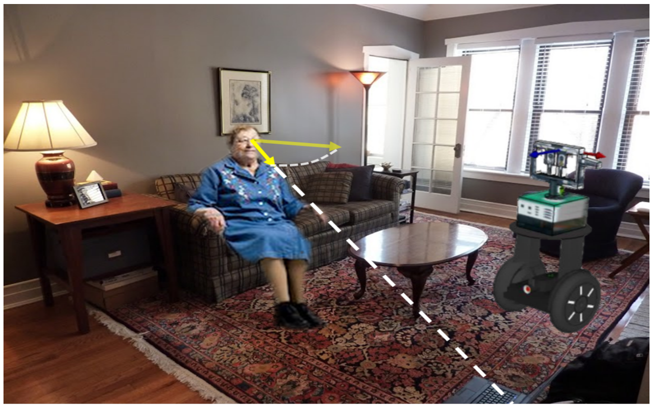


Fig. 1. Representational stage in joint attention. The artificial system should be able to build an internal representation of the environment and infer the object that the interlocutor is gazing during gaze following.

inside the social space and engages the robot meaning that he wants to initiate interaction. Afterwards she switches the focus of attention towards somewhere outside the field of view (e.g., a computer). In order to infer intention, and therefore construct the correct attentional behaviour, we need to discover the location or the object that the interlocutor is gazing because it could be important for the interaction being performed. In this case she could want to call by internet to her daughter.

In the study presented in [7], where an artificial attention system is tested without integrating the gaze interpretation ability, it is clear that the robot cannot infer the complete non-verbal message intended by the interlocutor. In that experiment, the interlocutor shows an object to the robot by deictic fixation (e.g. gazing or grabbing the object) and then the machine should search another object with similar characteristics. They show that just using the preatentive scene segmentation the robot is not able to understand the user needs, as the robot sometimes does not figure out which object is being showed. We hypothesise that enabling the machine with a full attentional system[1] will improve reciprocity, expectation fit and interaction [6]. This requires the implementation of the representational skill, which toddlers achieve in early stages of development [12], and involves discovering the object that the interlocutor is referring to. One of the possible approaches discussed in the literature is to provide an internal representation of the environment that works as a short-term memory and stores important information for attention [4, 8], and then use the gaze cue to modulate the potential attended objects.

[1] The majority of the works in attention lacks from the integration of all needed functionalities [3]. They usually use just visual sources, there is no top-down modulation nor emotional context and social signals are not taken into account.

In this paper we present a discrete log-spherical gaze tracing algorithm to be integrated in a full-fledged attention system [7] that establish a correspondence between a set of discretised cells with potential objects being fixated by the observer. We integrate the egocentric representation of the environment, Bayesian Volumetric Map (BVM) [4], with the gaze cue, improving the capacity of the machine to infer important regions even if they are outside the field of view and providing the robot with the representational stage, thus laying the foundation for non-verbal social HRI. Although the algorithm is focused on attention in HRI, it can be used as a general voxel traversal algorithm in bounded log-spherical space representations.

The paper is organised as follows: section 1.1 and 1.2 detail the motivation of this work and the current state-of-the art methods for gaze tracing; section 2 describes the proposed solution, formalizes it mathematically and presents the tracing algorithm; section 3 shows gaze tracing working, the computational analysis and the final results when integrating the BVM with the traversal algorithm in a real HRI scenario; finally, section 4 summarises the work.

1.1 Egocentric Representation and Gaze Tracing in Artificial Attention

Regarding to attention, the information provided by the different sensors must be subjected to a spatial correspondence [3]. For instance, a sound source should be related with its potential origin location in order to make possible the correspondent attention action [8]. In fact, in overt attention, where the scene is partially observed and changes depending on the actions (e.g., head movements), if we do not enable the machine to have an internal representation of the environment with temporal registration, its actions will become reactive for each location and angles setup. Furthermore, when placing the machines in a social or human interaction context, joint attention, a primal non-verbal communication process driven by attention, should be fulfilled [3, 6]. This mechanism, where an object or location of interest is shared just by engaging and deictic cues such as gaze, seems to be the backdrop for many social cognitive skills in humans [2]. For enabling this behaviour in machines, they must have their own spatial representation of the environment to correlate the gaze cues with the potential shared objects or locations. Therefore, any perception (e.g., visual, auditory, etc.), should be related and integrated into a single egocentric reference. Recent research works have introduced cluster [5] and spherical [13, 4] representations of egocentric space to deal with these issues. In this work we will use the 3D log-spherical representation proposed in [4] to codify the perceived environment. The method is also valid when the saliency of the scene is modelled [8].

The gaze cue provides interesting information about intention and it is crucial for social development [16, 3]. In humans' social interaction, when an interlocutor wants to share an object, he switches the focus of attention from the other's face towards the object. The other interlocutor starts a phase called gaze following [12, 2] to search the object or location that it is being shared. Therefore, first we need to estimate the interlocutor gaze direction and afterwards perform the gaze following until the object is discovered. Techniques to estimate and track the gaze are out of the scope of this paper. Nevertheless, an overview of recent approaches to infer gaze are described in [9]. For gaze following, apart from the naive and deterministic solution of redirecting the robot head according to the gaze vector, some developmental learning approaches have been researched [10].

1.2 Ray Tracing

Ray tracing in 3D has become very popular in games and simulators (e.g., PowerVR Ray or NVIDIA Iray) for light rendering and in medicine for image reconstruction [14, 15]. There are three key aspects for a good ray tracing algorithm: the computational speed, the generality to multiple inputs and the implementation simplicity. On one hand, we can find in the literature two optimal algorithms in the Cartesian space for different purposes. AmanatidesWoo's algorithm [1] is a general purpose method that computes the voxels that an arbitrary ray traverse in a bounded region (box). This algorithm computes the initial intersection point to speed up the computation. Siddon's Algorithm [14] uses ray tracing to optimize the reconstruction of medical images. Finally, Thibaudeau et al. [15] developed an algorithm that traces a ray using spherical or cylindrical coordinates. It analyses each dimension of the 3D space separately (radial, and azimuthal and elevation angles) and stores the entry and exit point of each voxel. Then the intersection points of all dimensions are sorted and the repetitions are eliminated.

2 Gaze Tracing in Log-spherical Coordinates

We generalize and abstract the problem as depicted in Figure 2. The interlocutor and the robot are represented by the green ball and the orange ball respectively. The gaze direction is described by the dashed black line and the log-spherical region is defined by the radius ρ and two angles (θ, ϕ). The gaze ray cuts through the bounded region starting in the intersection (green cross) and traverses a set of log-spherical voxels (represented in as red shapes in the figure) until it reaches the exit point. Note that further voxels from the robot are bigger because of the logarithm influence. Besides, ρ_{min} defines the egocentric gap [4]. Whilst this is the general case, the subject will usually be inside the spherical bounded region and therefore, the ray starting point will correspond to the gaze initial location[2].

2.1 Mathematical Formulation

We place the robot at the $(0, 0, 0)$ in the Cartesian coordinate system and define the perceptive space as a bounded spherical region by its radius ρ and azimuth

[2] The gaze initial location is commonly placed at the sellion: the point of the deepest depression of the nasal bones at the top of the nose.

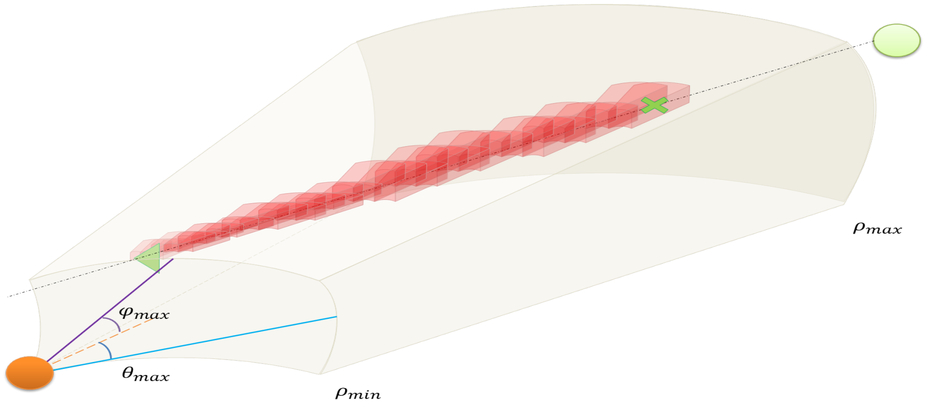


Fig. 2. Gaze tracing in a bounded log-spherical space partition. The region is compound of discrete voxels constrained by the minimum and maximum values of ρ, θ and ϕ.

θ and elevation ϕ angle ranges.

$$\rho_{min} \leq \rho \leq \rho_{max} \tag{1a}$$
$$\theta_{min} \leq \theta \leq \theta_{max} \tag{1b}$$
$$\phi_{min} \leq \phi \leq \phi_{max} \tag{1c}$$

The perceptive space of the robot is modelled by means of the BVM, an egocentric and metric inference grid that encodes the spatial occupation of the environment and its dynamics [4] as well as saliency [8]. The BVM space is defined in the log-spherical coordinate systems as follows,

$$\mathcal{Y} \equiv]log_b\rho_{min}; log_b\rho_{max}]\times]\theta_{min};\theta_{max}]\times]\phi_{min};\phi_{max}] \tag{2}$$

where the logarithmic base is,

$$b = a^{log_a(\rho_{max} - \rho min)/N}, \forall a \in \mathbb{R} \tag{3}$$

and N is the number of partitions in ρ dimension. The BVM is finally discretised into cells by defining the increment in the angles $\Delta\theta$ and $\Delta\phi$. Therefore, cell is indexed as C_{ijk} and represents the occupancy probability in the interval $[b^i + \rho_{min}, b^{i+1} + \rho_{min}] \times [\theta, \theta + \Delta\theta] \times [\phi, \phi + \Delta\phi]$. Note that i represents $\lceil log_b(\rho - \rho_{min}) \rceil$ (see Appendix). Furthermore, for a detailed explanation of the BVM see [4].

The gaze is represented as a ray expressed by the origin $p_0 = (x, y, z) \in \mathbb{R}^3$ and the director vector $\boldsymbol{v}_0$. Thus, given the parameter $t \in \mathbb{R}_0^+$ gaze is described by its Cartesian parametric equations,

$$y = p_0 + t\boldsymbol{v}_0 \tag{4}$$

The proposed method computes the collection of unique voxels traversed $\mathcal{C} = \{c_1, \ldots, c_n\}$ by the gaze ray inside the bounded region. With abuse of notation, the collection expressed in BVM indexes, as a matrix, is $\mathcal{C} = \{c_{i_1 j_1 k_1}, \ldots, c_{i_n j_n k_n}\}$.

2.2 Voxel Traversal Algorithm

Given an arbitrary gaze ray defined by Eq. (4) and the bounded region described by Eq. (1), we propose Algorithm 1 to compute the set of traversed voxels $\mathcal{C}$. The algorithm guaranties that $\mathcal{C}$ contains no repetitions and all voxels are inside the spherical region. First the logarithmic base is obtained using Eq. (3) and the initial intersection point between the ray and the region is computed as explained in section 2.3. Afterwards, we compute the next ray point using the default step. If the next ray point does not match the next voxel (i.e., the difference of indexes is not equal to 1) we apply binary search by reducing the step parameter dt to the half or incrementing by the half depending if we have fallen sort or long. In case of overshooting, when incrementing dt, in the next step we will arrive to the same cell, thus the middle point of the segment is computed. Finally, the new voxel found is added to the collection. The algorithm returns the collection $\mathcal{C}$ when the next point computed is out of bounds.

Algorithm 1 Log-spherical Gaze Tracing Algorithm

```
Require: $p_0, \boldsymbol{v}_0$    ▷ Ray origin and director vector
Require: $\rho_{min}, \rho_{max}, \theta_{min}, \theta_{max}, \phi_{min}, \phi_{max}$    ▷ Region bounds
Require: $N, defaultT$    ▷ Number of partitions of radius and initial increment

$\mathcal{C} = \emptyset$, $dt = defaultT$
$b = e^{\frac{\ln(\rho_{max} - \rho_{min})}{N}}$    ▷ Eq. (3)
$t_0 = initialT(p_0, v_0)$    ▷ See section 2.3
while inside do    ▷ Eq. (1)
    $\boldsymbol{p}_{xyz} = p_0 + t_0 \boldsymbol{v}_0$
    $\boldsymbol{p}_{ijk} = Cartesian2DiscreteLogSpherical(\boldsymbol{p}_{xyz}, b)$    ▷ See Appendix
    while $\neg found$ do
        $t_1 = t_0 + dt$
        $\boldsymbol{p'}_{xyz} = p_0 + t_1 \boldsymbol{v}_0$
        $\boldsymbol{p'}_{ijk} = Cartesian2DiscreteLogSpherical(\boldsymbol{p'}_{xyz}, b)$
        $\varDelta = |\boldsymbol{p}_{ijk} - \boldsymbol{p'}_{ijk}|$
        if $block$ then    ▷ overshoot or undershoot according to previous state
            $dt = 0.5(t_{prev} + t_1) - t_0$    ▷ the middle point is selected
        else if $\varDelta i > 1 \vee \varDelta j > 1 \vee \varDelta k > 1$ then
            $dt = 0.5dt$
        else if $\varDelta i = 0 \wedge \varDelta j = 0 \wedge \varDelta k = 0$ then
            $dt = 1.5dt$
        else
            $found = true$, $t_0 = t_1$, $dt = defaultT$
            $\mathcal{C} = \mathcal{C} \cup C_{ijk}$    ▷ Add new voxel to the set
        end if
    end while
end while
return $\mathcal{C}$
```

The complexity of the algorithm is in worst case $O(n \log m)$ due to the inner loop that uses binary search. In practice the number of iterations m that binary search is processed for each outer iteration is quite smaller than n and depends on the $defaultT$ parameter used. This means that the method theoretically is faster than any optimal sorting algorithm, but again, in practice this approach depends on the function $Cartesian2DiscreteLogSpherical$ that transforms the Cartesian point into the discrete log-spherical coordinates (i.e., BVM matrix indexes). The algorithm is generalizable as it works for any kind of discrete representation just by substituting the conversion function.

2.3 Computing the Initial Intersection Point

We can rewrite the spherical region defined by inequalities (1a),(1b),(1c) into Cartesian coordinates by converting the spherical parameters (ρ, θ, ϕ) into the Cartesian system following the equations defined in (8) (see appendix). The new inequalities become:

$$\rho_{min} \leq \sqrt{x^2 + y^2 + z^2} \leq \rho_{max} \tag{5}$$

$$\theta_{min} \leq \arctan\left(\frac{x}{z}\right) \leq \theta_{max} \tag{6}$$

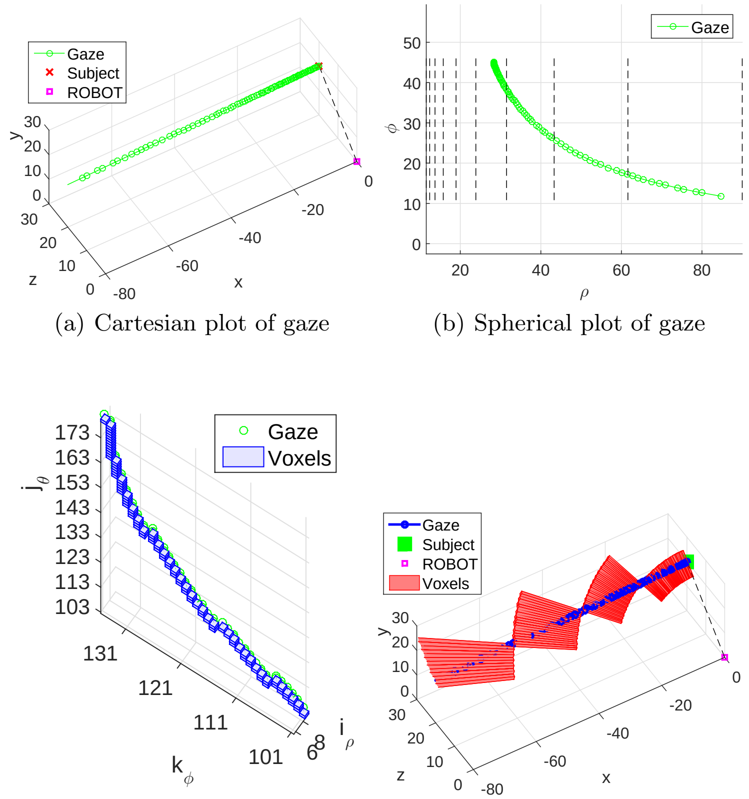
$$\phi_{min} \leq \arctan\left(\frac{y}{\sqrt{x^2 + y^2}}\right) \leq \phi_{max} \tag{7}$$

Then we substitute x, y, and z with Eq. (4), and solve for t as it is detailed in the appendix. The minimum t value that satisfies the equations defines unequivocally the initial intersection point t_0.

3 Results

3.1 Gaze tracing algorithm example

By means of a prepared example we show the algorithm functioning. The robot is stationary at $(0, 0, 0)$ and the gaze is defined by its initial point $p_0 = (0, 20, 20)$ and its direction $\boldsymbol{v}_0 = (-0.3, -0.01, 0)$. The bounded spherical region is constrained to: $\rho \in (10, 80)$, $\theta \in (-180, 180)$ and $\phi \in (-90, 90)$. Figure 3(a) shows in Cartesian coordinates the abstracted scene where the subject is gazing down and left. Figure 3(b) shows the same scene in spherical coordinates (i.e., ϕ and ρ plane). The dashed lines describe the separation between cells in the ρ axis defined by $b^i + \rho_{min}$ in the logarithmic space. We can see that the distance between the boundaries increase exponentially making closer regions to be more fine grained. The output of the algorithm, the set of traversed voxels, is shown in Fig. 3(c), where the indexes in all three dimensions are plotted as boxes. In this presented sub region of the BVM matrix (i.e., discretised log-spherical structure) we can see that as the ray goes further from the robot θ angle is increased resulting in a j decrement. Note that the angles values are positive as they are the



(a) Cartesian plot of gaze

(b) Spherical plot of gaze

(c) Voxels indexes of BVM matrix

(d) Traced voxels in Cartesian space

Fig. 3. Gaze tracing example. (a) Cartesian plot with a subject looking left and down; (b) Gaze ray converted to spherical; (c) Output of the algorithm: set of traversed voxels and their indexes in the BVM matrix; (d) Computed voxels transformed into Cartesian space overlay figure (a). Note that we represent Cartesian units in decimeters.

indexes of the matrix. Finally Fig. 3(d) shows the voxels defined by the indexes converted into the Cartesian coordinate system. The egocentric nature of the log-spherical representation make the red trapezoids that represent the voxels to face the robot location.

3.2 Computational Results

In order to analyse the computational complexity of the proposed algorithm we compare it with two other state-of-the-art voxel traversal algorithms: AmanatidesWoo that works in the Cartesian space [1] and Thibaudeau that computes the intersections in the spherical space [15]. The comparison is performed by modifying the number of voxels that must be traversed. All implementations have been developed in Matlab and have been tested on an Intel Core i7-4700MQ CPU computer with 8Gb of RAM. To reduce the noise due to tasks latency we have run the algorithms 200 times for each number of voxels value.

Figure 4 shows the time comparison in ms of the three algorithms. We can see that AmanatidesWoo wins but it is constrained to Cartesian configurations that can exploit the mathematical properties of 3D rectangles. Although our algorithm behaves similar to Thibaudeau in terms of time computation works for log-spherical spaces, it is simpler and more generalizable. In contrast, our algorithm requires experimental tuning of the $defaulT$ parameter to minimize the computation time. For the statistical analysis we have set it to 1.5531 metres.

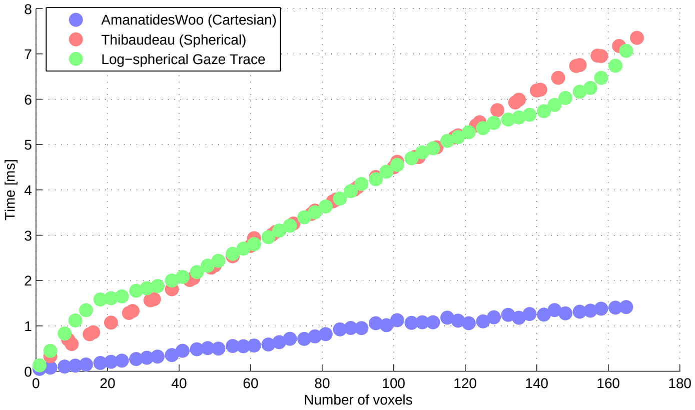
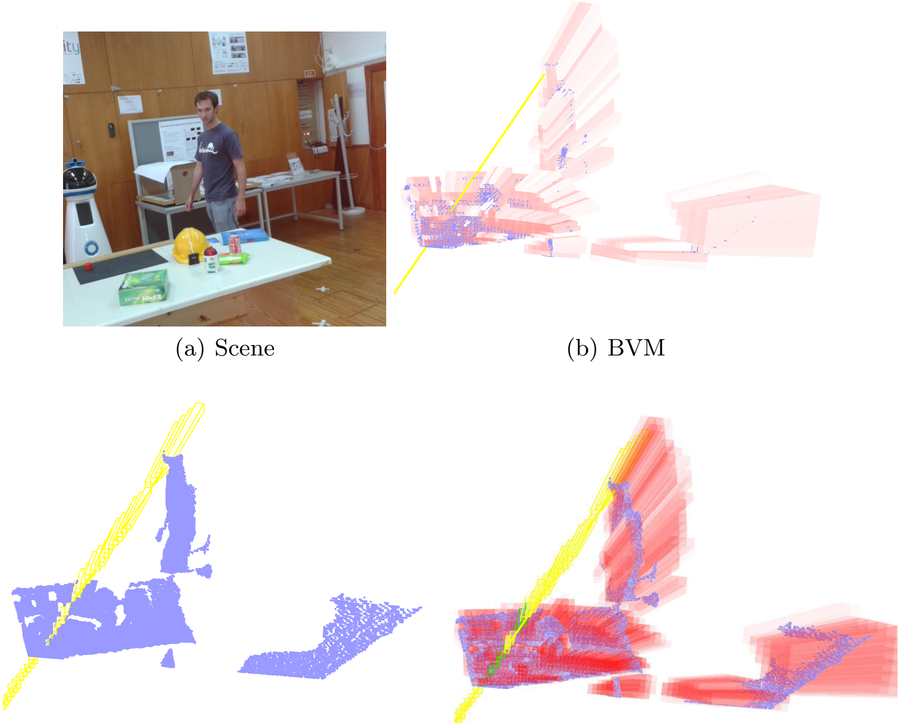


Fig. 4. Computational time comparison depending on the number of voxels traversed.

3.3 Integrating the Gaze tracing with the 3D occupancy representation

We combine the gaze tracing proposed algorithm with the log-spherical occupancy grid (BVM) to achieve our final goal that is to discern the potential objects that the interlocutor is staring at. The BVM stores as a short-term memory the probability of occupied regions. Thus, the gaze traversed voxels that intersects the occupied ones are the locations of interest. We define the experimental setup, as depicted in Fig. 5(a), with a subject that enters in the field of view of the robot and looks to an object placed on the table. The wanted voxels must satisfy that are traversed by the gaze ray and are occupied. We define an occupied voxel when its probability is higher than 0.6. The gaze tracing algorithm output is shown in Fig. 5(c) and the final pursued intersected locations are plotted as green trapezoids in Fig. 5(d). We can see that the voxels are correctly found on the table.



(a) Scene

(b) BVM

(c) Gaze tracing

(d) BVM-Gaze intersection

Fig. 5. Gaze trace combined with the BVM. (a) Robot camera; (b) Occupancy log-spherical grid (red trapezoids) with the overlaid gaze ray (yellow line); (c) voxels traversed by the gaze in the log-spherical space (yellow trapezoids); (d) Intersected voxels between the gaze ray and the occupancy grid (green trapezoids). Note that the red intensity determines the probability of a voxel being occupied and that the point cloud is overlaid in blue in (b), (c) and (d).

4 Conclusion

The penetration of social robots in assisted living applications is slowly occurring mostly due to the incapability of deploying machines with enough coherent behaviour according to human expectations. The presented results have shown how easy the proposed approach is integrated into an artificial attention system in order to identify the potential objects of interest. Thus, this proposal, as it allows to combine a social signal (gaze) and the short term-memory (egocentric representation of the environment and some stored important objects), will enable machines to understand other's attention based on his deictic gaze fixations. Furthermore, it will help the robot to recognise where the interlocutor is attending, improve subject's gaze following and enhance reciprocity and non-verbal social interplay.

We have presented a fast voxel traversal algorithm for gaze tracing in discretised log-spherical bounded spaces that works in real time. Its worst case complexity is $O(n \log m)$ and m depends on a parameter that can be experimentally optimized. The comparison of other state-of-the-art algorithms shows that the proposed approach, which is simple to implement, behaves as fast as them but allowing generalization to any bounded discretised space configuration.

When representing the subject's gaze with just one ray the uncertainty and error associated with the gaze detection measurement is not taken into account. For that purpose, we will extend the algorithm to compute a cone of rays, whose parameters will model the gaze uncertainty. Moreover, we will look for optimized implementations of the algorithm by including some of the nice properties of [15]. The extended implementation of the algorithm will be embedded in a full artificial attentional mechanism to evaluate if artificial systems can be improved when enabling the attention skill [6].

5 Appendix

Conversion from Cartesian to spherical coordinates.

$$\rho = \sqrt{x^2 + y^2 + z^2} \qquad \phi = \arctan\left(\frac{y}{\sqrt{x^2 + y^2}}\right) \qquad \theta = \arctan\left(\frac{x}{z}\right) \tag{8}$$

Conversion from Cartesian to discretised log-spherical. Applying Eq. (8), and the following equations, the discretised log-spherical indexes are found.

$$i = ceil(\log_b(\rho - \rho_{min})) \qquad j = floor(\phi) - \phi_{min} + 1 \qquad k = floor(\theta) - \phi_{min} + 1$$

Computing the initial intersection point. Substituting the ray equation (4) into the inequalities presented in (5), (6) and (7) and solving for the parameter t we obtain the first intersection point of the gaze ray with the bounded spherical region.

$$t_\rho = -\frac{(x_0v_x + y_0v_y + z_0v_z)}{v_x^2v_y^2v_z^2} \mp \mp \sqrt{\left(\frac{(x_0v_x + y_0v_y + z_0v_z)}{v_x^2v_y^2v_z^2}\right)^2 - \frac{{x_0}^2 + {y_0}^2 + {z_0}^2 - {\rho_{min}}^2}{v_x^2v_y^2v_z^2}} \tag{9a}$$

$$t_\phi = -\frac{(x_0v_x + z_0v_z - y_0v_y\tan(\rho)^2)}{v_x^2 + v_z^2 - v_y^2\tan(\rho)^2} \pm \pm \sqrt{\frac{0.5(x_0v_x + z_0 \times v_z - y_0v_y\tan(\rho)^2)^2}{v_x^2 + v_z^2 - v_y^2\tan(\rho)^2} - \frac{{x_0}^2 + {z_0}^2 - {y_0}^2\tan(\rho)^2}{v_x^2 + v_z^2 - v_y^2\tan(\rho)^2}} \tag{9b}$$

$$t_\theta = \frac{x_0\tan(\theta) - z_0}{v_z - v_x\tan(\theta)} \tag{9c}$$

Each resultant t_ρ, t_ϕ, t_θ will be the minimum value calculated using each bound. The final t is computed by overlapping the intervals and extracting the closest one to the gaze ray origin that satisfy the conditions.

References

1. Amanatides, J., Woo, A., et al.: A fast voxel traversal algorithm for ray tracing. In: Eurographics. vol. 87, pp. 3–10 (1987)
2. Brooks, R., Meltzoff, A.N.: The development of gaze following and its relation to language. Developmental science 8(6), 535–543 (2005)
3. Ferreira, J.F., Dias, J.: Attentional Mechanisms for Socially Interactive Robots – A Survey. IEEE Transactions on Autonomous Mental Development 6(2), 110–125 (2014)
4. Ferreira, J.F., Lobo, J., Bessire, P., Castelo-Branco, M., Dias, J.: A Bayesian Framework for Active Artificial Perception. IEEE Transactions on Cybernetics (Systems Man and Cybernetics, part B) 43(2), 699–711 (April 2013)
5. Kuhn, B., Schauerte, B., Kroschel, K., Stiefelhagen, R.: Multimodal saliency-based attention: A lazy robot's approach. In: Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. pp. 807–814. IEEE (2012)
6. Lanillos, P., Ferreira, J.F., Dias, J.: Evaluating the Influence of Automatic Attentional Mechanisms in Human-Robot Interaction. In: Workshop: a bridge between Robotics and Neuroscience Workshop in Human-Robot Interaction, 9th ACM/IEEE International Conference on. pp. 1–2. Bielefeld, Germany (March 2014)
7. Lanillos, P., Ferreira, J.F., Dias, J.: Designing an Artificial Attention System for Social Robots. In: Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on. IEEE (2015), to appear
8. Lanillos, P., Ferreira, J.F., Dias, J.: Multisensory 3D Saliency for Artificial Attention Systems. In: 3rd Workshop on Recognition and Action for Scene Understanding (REACTS), 16th International Conference of Computer Analysis of Images and Patterns (CAIP). pp. 1–6 (2015)
9. Murphy-Chutorian, E., Trivedi, M.M.: Head pose estimation in computer vision: A survey. Pattern Analysis and Machine Intelligence, IEEE Transactions on 31(4), 607–626 (2009)
10. Nagai, Y.: Joint attention development in infant-like robot based on head movement imitation. In: Proc. Third Int. Symposium on Imitation in Animals and Artifacts (AISB05). pp. 87–96. Citeseer (2005)
11. Santos, L., Christophorou, C., Christodoulou, E., Samaras, G., Dias, J.: Development strategy of an architecture for e-health personalised service robots. IADIS International Journal On Computer Science and Information Systems 9, 1–18 (2014)
12. Scassellati, B.: Theory of mind for a humanoid robot. Autonomous Robots 12(1), 13–24 (2002)
13. Schillaci, G., Bodiroža, S., Hafner, V.V.: Evaluating the effect of saliency detection and attention manipulation in human-robot interaction. International Journal of Social Robotics 5(1), 139–152 (2013)
14. Siddon, R.L.: Fast calculation of the exact radiological path for a three-dimensional CT array. Medical Physics 12(2), 252–255 (1985)
15. Thibaudeau, C., Leroux, J.D., Fontaine, R., Lecomte, R.: Fully 3D iterative CT reconstruction using polar coordinates. Medical Physics 40(11), 111904 (2013)
16. Vernon, D., von Hofsten, C., Fadiga, L.: A roadmap for cognitive development in humanoid robots, vol. 11. Springer (2010)
17. Vinciarelli, A., Pantic, M., Heylen, D., Pelachaud, C., Poggi, I., D'Errico, F., Schroeder, M.: Bridging the gap between social animal and unsocial machine: A survey of social signal processing. Affective Computing, IEEE Transactions on 3(1), 69–87 (Jan 2012)