Context-based understanding of interaction intentions

João Quintas[1,2] Luis Almeida[1,3] Miguel Brito[1] Gustavo Quintela[1] Paulo Menezes[1] Jorge Dias[1,4]

Abstract— **This paper focus in the importance of context awareness and intention understanding capabilities in modern robots when faced with different situations. The inclusion of such requirements in robot design aim for more intelligent robots capable to adapt its behaviours to the faced situations. Gaze estimation and gesture interpretation are modalities, closely related with context-depent human intention understanding, that are addressed in this work.**

I. Introduction

It is common-sense that humans' and other animals' behaviours are context-dependent. Context affects almost all aspects of behaviour, mostly in an automatic manner,i.e. without a conscious reasoning effort.

Human perception is heavily influenced by top-down predictions, making it more difficult to detect, or recognize, out-of-context objects than familiar ones, and there are numerous studies showing the priming effect of one concept on another (see e.g. Glass and Holyoak, 1986 [1]). Pevtzow and Goldstone (1994) [2] suggest that the categories a person has learnt, affects which features of an object he or she perceives. Context has also impact on decision-making and action. Preference measurements are also context-sensitive [3]. Learning is affected by context, as studies as far back as those of Pavlov have demonstrated. Context has been studied extensively in language use, usually with "context" meaning the history of prior utterances (e.g. Ferstl, 1994 [4]), but also including other kinds of context. Holtgraves [5] has found that the status of the speaker relative to the hearer affects whether the literal meaning of an indirect request is activated, or not.

According to Turner [6], we can define context and situation as follows. The term context means any identifiable configuration of environmental, mission-related and agent-related features that has predictive power for an agent's behaviour. The term situation is used to refer to the entire set of circumstances surrounding an agent, including the agent's own internal state. Context is thus the elements of the situation that should impact behaviour.

Artificial intelligent agents, as personal robots, must be context-sensitive and adapt its own behaviour. The context-mediated behavior (CMB), presented in [6], is based on the idea that an agent should have explicit knowledge about contexts in which it may find itself, then use that knowledge when in those contexts.

In human interaction, the contextual information can influence the understanding of personal intention. We can infer human intention by analysing specific types of features in a person's pose and combining them with the context to understand the whole situation. In spite of context, other types of information are needed to provide adequate interactivity. In this work we address the gaze orientation to infer the person's visual attention.

Visual attention is associated with person eye fixations [7]. By fixating our attention to the same point where another person is staring, we emphasise our presence through a shared attention process [7]. Levels of attention associated to gaze directionality has been subject of study in the Theory of Mind [8]. While performing a task, user's eyes and/or head directionality patterns can be used to extract measures of the level of attention. For example, in human-robot interaction (HRI), by fixating on specific robot points (screen points, body parts or camera) or watching away those interest regions, the user demonstrates his interest or attention to the robots' behaviour. The link between attention and gaze movements was demonstrated by several authors [9][10]

In what concerns gestural interaction, humans use gestures to indicate directions, places or objects, repetitive gestures to "control" actions or call for attention, and expression-related or communicative gestures. Many authors have devoted their efforts to gesture recognition in the past decade, especially in attempts to use sign language like methods to communicate with robots [11]To our knowledge, only a few researchers became interested to deictic[12] and periodic[13][14] gestures in recent years. These two types of gestures are of large importance to HRI, as pointing directions, places, or objects can simplify largely the communication between humans and robots. Repetitive or periodic gestures can also provide a simple, but natural and effective way of communicating with a robot. Among the repetitive gesture class we can include: left (moving hand horizontally left-and-right, but faster to the left), right, slow down, approach, etc. These types of gestures can have a great utility in industrial plants, where transportation robots may be guided by any person along a new path for some reason.

The next sections will present a review of the literature that was relevant to guide our studies, some methodology considerations, the implementation and results, attained at the current stage of the work, and finally some conclusions for this paper.

[1] Institute of Systems and Robotics - University of Coimbra, Coimbra, Portugal {jquintas, laa, mbrito, gquintela, paulo, jorge}@isr.uc.pt

[2] Institute Pedro Nunes, Coimbra, Portugal

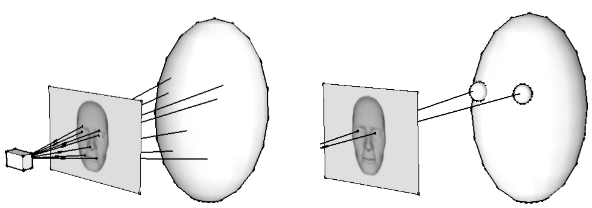
[3] Instituto Politécnico de Tomar, Tomar, Portugal

[4] on sabbatical leave at Khalifa University of Science, Technology and Research (KUSTAR), Abu Dhabi, UAE.

II. METHODOLOGY

A. Time varying analysis of eye/head gaze movement

Retrieving gaze estimation from images, or a movie is something that is natural for most humans. For a robot to have that capability is not as trivial. For gaze estimation we need four points, which are the two irides centres and the two eyeballs centers; the two vectors that we get from projecting a ray that passes through the eyeballs centers and corresponding iris will intersect in space, in the point that a person is looking at. Having this, we will have to recognize the iris, locate the eyeballs centers, and finally trace the two vectors to estimate the gaze. Using an RGB camera, first step is to locate the user's face, using Viola and Jones method [15] for face detection. From this detection, and using an average ellipsoid model of the user's head, we can get a rough estimate of the position of user's face w.r.t. the camera. The next step consists in detecting robust image features in the user's face and map them onto the ellipsoid model. We have chosen SURF [16] as the feature detector, and for each feature we compute the respective projective ray with the model to get the corresponding 3D points, as in figure 1a.



(a) (b)

Fig. 1: (a) Representation of projection of SURF points to the ellipsoid model and b Locating eyeballs in the ellipsoid model.

Now we need a eye tracking method to get the both irides' centers. Once we have them we can map them onto the ellipsoid model. Human eyes are approximately spherical with 22.5mm of diameter, thus we can approximate their centers in the ellipsoid frame (Fig. 1b).

From this point on, the system detects the user's face, and extracts new SURF points. These new SURF features are paired with the corresponding obtained in the first frame. The head pose estimation corresponds to the application of one method to solve this PnP problem. The result is a approximation of the user's head pose and as so, the position of the eyeballs centers. Again using the eye tracking we get the iris, and ray trace them to the spheres models of both eyes. Finally we have the four 3D points we needed for gaze estimation, the eyeballs centers and the iris. The two vectors we obtain intersect in space at the location the user is looking at. The eye tracking routine is made using OpenCV functions. From one image of the user's face, that we get from the HAAR cascade for face detection, we heuristically narrowed the region of the eyes.

B. Periodic gestures

A periodic gesture can be defined as the repetitive motion of a person's hand and that, depending on the envelope or direction, means hello, move right, move forward, among others. This class of gestures are mainly related to action guidance, or to draw attention or compliance. By observation we have concludes that these gestures are normally performed, by different people, in a range of frequencies between the 0.5Hz and 1.5 Hz. By consequence to detect these gestures we need to detect the presence of a periodic trajectory of the hand within the given range of frequencies.

A similar problem that we can find in the literature is the detection of voice signals in audio streams. Both the periodic gestures and in case of detecting voice, these have a natural frequency and a certain amplitude. In addition, both are normally corrupted with noise, that needs to be eliminated so that the detection can take place more easily. Given these factors, we used an adaptation of a voice detection algorithm namely the *Modified Autocorrelation Method Using Clipping* (AUTOC) [17]. This algorithm has a quite simple principle of operation,that had to be adapted to the current problem. This was due to the fact that we are dealing with a range of very low frequencies and sequences with a small number of samples, when compared to the voice signals.

In Fig. 2 represents a block diagram showing the steps of this algorithm. In each 120 samples ($S[n],\ n=1,\cdots,120$), the signal is submitted to a low pass digital filter (FIR filter of order 5) with a cutoff frequency of 10Hz in order to eliminate the highest frequency noise, taking into account the range of the gesture working frequencies of the gestures. Next, it was necessary to rectify the amplitude values, and eliminate the existing offset so that clipping is done in a form both cohesive and uniform. To remove the offset, we compute S_{max} (1) and S_{min} (2) and using the expression (3) the data is rectified.

$$S_{max} = max(S[n]), n = 1,\cdots,120 \quad (1)$$

$$S_{min} = min(S[n]), n = 1,\cdots,120 \quad (2)$$

$$S_c[n] = S[n] - \frac{S_{min}+S_{max}}{2}, n = 1,\cdots,120 \quad (3)$$

The next step is to do the clipping by the level cL for each sequence of 120 samples. After computing the maximum values for both the first, and last 20 samples, respectively IPK1, and IPK2. Them cL is determined as $cL = k \times \min(IPK1, IPK2)$, where $k = 0.64$. Using this clipping level, a centred sequence of values (see figure 3b)) is generated from the input signal as follows: 1 if the value $S_c[n]$ is greater than cL, -1 if the value of $S_c[n]$ is less than cL and 0 for all other cases [17].

The normalised autocorrelation of the centred sequence is obtained, and the resulting sequence corresponding to the different lag values is scanned for the first minimum. If this minimum value is below -0.35, then the signal represented by the 120 samples is considered as periodic, otherwise it is labelled as non-periodic. It is important to note that this is in fact different from the AUTOC algorithm, whereas

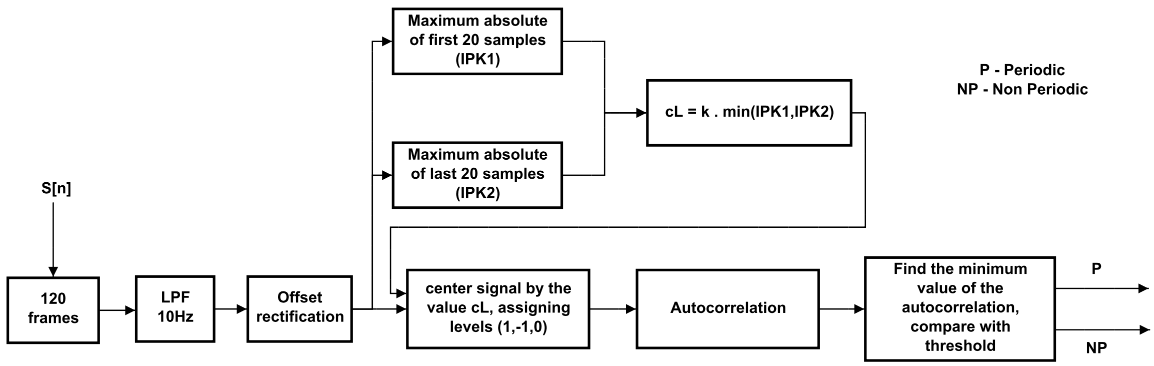


Fig. 2: Block diagram for detection of periodic gesture algorithm

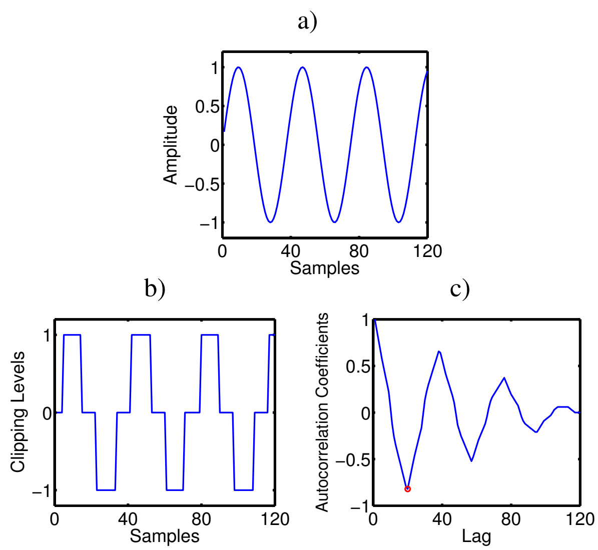


Fig. 3: Some algorithm steps submitted to a well know sinusoidal signal: a) well know sinusoidal signal $x(n)$; b) clipping step; c) autocorrelation result with threshold=-0.35.

the in latter it is considered the first maximum value of the normalised autocorrelation. In reality the first minimum value, whose lag corresponds to halt the signal period (that results in the phase opposition of the signal), whereas the first maximum value corresponds to the lag equivalent of the period of the signal. We have chosen the minimum as it is more easy to detect especially with such small number of samples.

Now our system is able to detect if a periodic gesture is present or not, and we can select the principal direction as well as its intensity to be analysed in order to identify the associated semantic.

C. Deictic Gestures

Among the group of deictic gestures we are mainly interested in the pointing gestures. Using these gestures we can interact with a robot specifying a position to which it should move, or an object it should manipulate, or even a person it should interact with.

The approach used here is not to determine the pointing gesture alone. Using first a attention gesture (waving, for example), to get the attention of the robot, we can then point to a certain place for it to move. Using a depth camera and the application programming interface OpenNI [18], [19] it is possible to get the positions of the elbow and the hand. This pair of 3D points define the pointing vector $\mathbf{a}$. To determine the position the user is intending the robot to go, we make use of the ground plane information given by the OpenNI. The ground plane is then defined by a normal vector $\mathbf{n}$ and 3D point $\mathbf{p_0}$ of the floor as:

$$\mathbf{n}\cdot(\mathbf{p}-\mathbf{p_0}) = \mathbf{0}, \tag{4}$$

knowing that $\mathbf{p}$ is a generic point on the plane.

With the vector $\mathbf{a}$ and the plane, defined in (4), we can know the intersection point, which should correspond to the position indicated by the user. Using the vector equation for a line and the equation of the plane (4),

$$\begin{cases} \mathbf{n}\cdot(\mathbf{p}-\mathbf{p_0}) = \mathbf{0} \\ \mathbf{p} = \mathbf{l} + \lambda\cdot\mathbf{a}, \end{cases} \tag{5}$$

defining $\mathbf{l}$ as a point in the line, with the same direction as $\mathbf{a}$, and λ relates $||\mathbf{a}||$ and the distance between the $\mathbf{l}$ and the point of the plane, $\mathbf{p}$. Determining the equation system (5) we can define the λ as,

$$\lambda = \frac{(\mathbf{p}-\mathbf{l})\cdot\mathbf{n}}{\mathbf{a}\cdot\mathbf{n}}.$$

Enviroment plane segmentation: The environment that surrounds the robot must be known in order to distinguish the pointing gesture according the context. For example, if direction pointing line intersects a tall structure probably it might refers an object to grab (ex: on a table), otherwise might just refer a location where the robot must go. A rough environment segmentation method is suggested based on 3D Hough plane detection [20]. It is well known that a plane can be defined through the following equation:

$$\begin{aligned} \rho &= \mathbf{p}\cdot\mathbf{n} \\ &= p_x n_x + p_y n_y + p_z n_z \end{aligned}$$

with $\mathbf{p}$ belonging to the plane, $\mathbf{n}$ being a unitary normal vector and ρ the distance to the origin. Considering the plane polar coordinates representation the equation can be rewriten as:

$$\rho = p_x \sin\varphi \cos\theta + p_y \sin\varphi \sin\theta + p_z cos\varphi$$

where φ is the angle between xy-plane and the normal vector in z direction, and θ is the angle of normal vector projection on xy-plane. A $\mathbb{R}^3$ plane will be mapped on a tridimensional Hough Space (θ, φ, ρ). For each Cartesian point from a 3D cloud, we mark all the planes that include it on this Hough Space. The intersection of three curves represents a plane. Finding where higher intersections happens, it is possible to detect dominant planes through a threshold (see algorithm 1).

Algorithm 1 3D Hough Transform Plane Detection

```
Input: 3D point cloud
Output: dominant planes
for all points p from 3D point cloud do
    for all cells (θ, φ, ρ) in accumulator HT do
        if point p is inside the plane (θ, φ, ρ) then
            increment cell HT(θ, φ, ρ)
        end if
    end for
end for
find HT(θ, φ, ρ) cells with higher values, that defines
    dominant planes
```

III. Implementation and Results

A. Gaze tracking

The head pose estimation with gaze tracking is implemented in C, in a normal laptop and using it's own webcam. This application runs in real time at about 20 frames per second. The head pose estimation with gaze tracking is good for a normal distance of interaction of about 50 to 80 cm, which for a longer distance, the rate of correct matched points (matches between the reference frame and other frame) is reduced and the head pose estimation is no longer possible. The pose estimation has a fluctuation error of about 4 mm measured using a human dummy. The eye tracker has an error of about 10% of the dimension of the detected iris, proven by visual marking of iris centre and result of eye tracker (Fig. 4), for a distance no larger than 1.5m, in which the irides become too small to be detected. Combining the head pose estimation with the eye tracker we end up with a estimation of the user's gaze. The gaze estimation has an error of about 3 degrees, but in our application is enough for detecting the region which the user is looking at, and so, we extrapolate the user's attention. If the user is looking to the region of the robot, then we may consider the he/her is paying attention to the robot.

The gaze tracking system based on head pose estimation enable to limit regions and determine when the user is looking at them, or when is not looking. Such functionality on a robot (Fig. 5) provides an additional input, giving the robot the ability to answer the question ”Is the user looking at me?”.

Fig. 4: Example of eye's image filtering and estimation of iris center

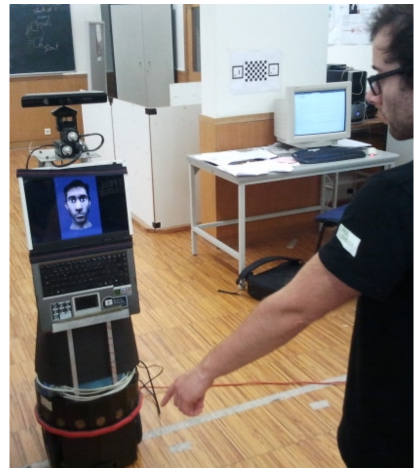


Fig. 5: Hilario, a human interaction robot used for context-based understanding experiments: context-sensitive for periodic and deictic gestures and attention based on user gaze.

B. Detection of periodic gestures

In Fig. 6 is shown the three coordinates of the periodic move forward gesture. With its analysis we can conclude that the implementation can perfectly distinguish the coordinates that make the gesture periodical with those that do not. Since the principal component of this gesture is on coordinate Z axis, we can expect that the periodicity is detected primarily on the corresponding coordinate values. If fact, as long as we can detect that there is a periodic behaviour in one of the coordinate axes we immediately classify the gesture as periodic. In addition, the implementation has a time response of detection between 3 and 4 seconds, which corresponds to the number of samples that the analysis is based upon.

C. Pointing Gesture

In what concerns, pointing gestures, as it is illustrated in Fig. 7a, we are able to indicate a position to the robot that it should go to, what corresponds to identify a point on the ground floor. We were also able to measure the precision of the system comparing the distance between a know marked point and the samples average, taking the Primesense sensor as origin. As show in Fig. 7b we can conclude that the system, in order to tell the robot to go to specific area, have a good precision results. We are currently implementing the object plane segmentation that will enable us to identify pointed objects using a similar method.

The object plane segmentation enable us to identify pointed objects using a similar method. The plane detection approach described in algorithm 1 identifies dominant planes by searching high peaks of sinusoidal surfaces in-

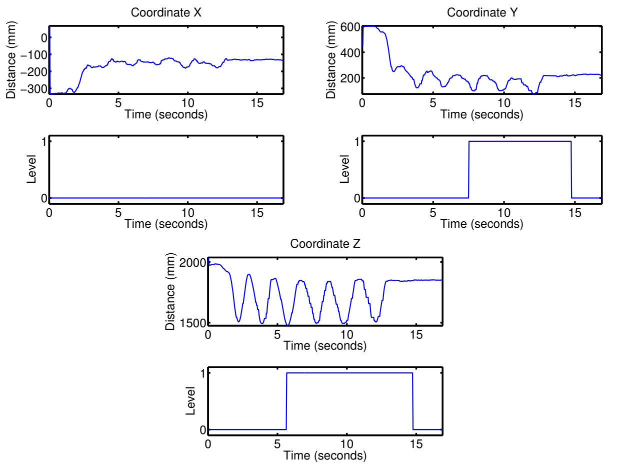
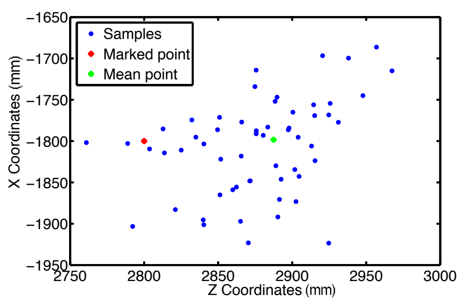


Fig. 6: The three coordinates of the move forward gesture and the corresponding result of periodic the characteristics.

(a)



(b)

Fig. 7: (a) Result of pointing gesture. The white point is the location where the gesture is directed.; (b) Results obtained with the measurement of precision, comparing the mean point of 120 samples and the marked point at 2.8m depth and 1.8m to the left of the sensor reference.

tersections on Hough Space (θ, φ, ρ). Figure 8 exemplifies the intersection of 3 sinusoidal surfaces generated by 3 three-dimensional points (a horizontal plane defined by (1,0,1),(0,1,1),(1,1,1)).

D. Context representation and identification

The context was described using an ontology designed using Protégé. The ontological representation for context is depicted in figure 9.

This ontology provides a mechanism to relate all the relevant information that contextualize a given gesture. Extracted features are characterized in terms of mission, environment and temporal characteristics. The interpretation of a gesture takes into account the relationships between these characteristics, providing the context for that gesture.

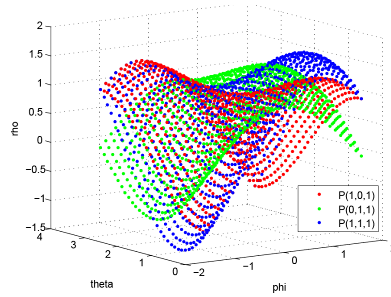


Fig. 8: Three 3D points mapped on Hough Space (θ, φ, ρ). The intersection of the 3 sinusoidal surfaces locks the plane defined by the three 3D points.

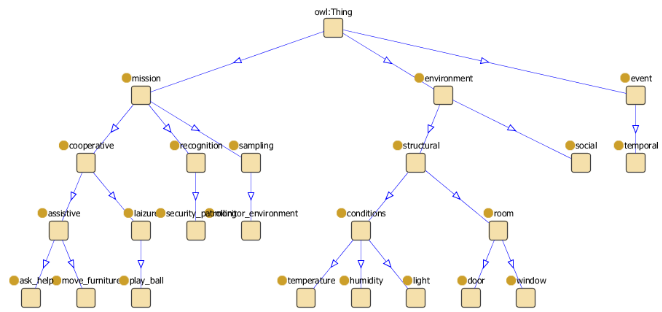


Fig. 9: Example of a possible context descriptor using an ontological representation

The context identification problem was considered to be similar to an ontology matching problem. This means, given a set of input features, the process selects the most similar representation for context, according an evaluation metric. The matching process was implemented using a genetic-algorithm (GA), implemented with the WatchMaker library. The fitness function uses an evaluation metric based in the Tversky's similarity [21]. The ratio model we applied is modeled according to the equation:

$$Similarity_{O_1,O_2}(M) = \frac{f((F_{O_1} \in F_{O_2})|M)}{f((F_{O_1} \in F_{O_2})|M) + \alpha \times f((F_{O_1} - F_{O_2})|M) + \beta \times f((F_{O_2} - F_{O_1})|M)} \tag{6}$$

where $f((F_{O_1} F_{O_2})|M)$ are the matched elements of both ontologies with respect to the mapping M, $f((F_{O_1} - F_{O_2})|M)$ and $f((F_{O_2} - F_{O_1})|M)$ are respectively two sets of the unmatched elements with respect to the mapping M. α and β are two parameters between 0 and 1, which determine the relative importance of the two unmatched feature sets; f is a function defined as the cardinality of set.

In our genetic-algorithm, the Elitist strategy was employed to save the current 5% of best solution after selection, i.e. 5% of the population with best fitness score was passed to the next generation unchanged. The parameters used in the genetic algorithm were assigned as follows: size of population was 1000; crossover points were 3; crossover probability was 0.9; mutation probability was 0.001; and the max generation count was 400. The genetic algorithm parameters were determined heuristically, thus more advanced techniques for tuning these values will be studied as future work.

The candidate solutions were represented as an array of integer numbers obtained by a stochastic process. The search space for a given solution was limited to the number of concepts of both ontologies, i.e. the length of the array was given by the number of concepts in the ontology 1 whilst the value for each array "cell" could vary between 0 and the number of concepts in ontology 2. Each concept in one ontology could be mapped in more than one concept in the other, thus value repetition throughout the candidate solution were allowed. The candidate solution representation used in our study is similar to that used in [22], considering the adequate adaptations to each algorithm. For the genetic algorithm, given its population characteristics, a number of candidate solutions were computed at the beginning, according the population size, and then changed according the mutation and cross-over parameters.

The ontology matching algorithm was tested in the OAEI2005 dataset. The evaluation measures used were respectively Precision (Pre), Recall (Rec) and FMeasure(F1). The obtained results are summarized in table 10 and depicted, comparing different approaches to the problem.

Measure	Benchmark	Falcon	Dublin20	Foam	GAOM	Genetic (o)	SimAnneal (o)	Genetic (n.o)	SimAnneal (n.o)
Pre	1xx	1,00	1,00	0,98	1,00	0,72	0,40	0,96	0,53
	2xx	0,90	0,94	0,89	0,92	0,75	0,47	0,75	0,47
	3xx	0,93	0,67	0,92	0,89	0,15	0,06	0,59	0,25
Rec	1xx	1,00	0,99	0,65	1,00	0,71	0,43	0,94	0,57
	2xx	0,89	0,71	0,69	0,80	0,51	0,34	0,51	0,35
	3xx	0,83	0,60	0,69	0,82	0,09	0,05	0,28	0,15
F1	1xx	1,00	0,99	0,78	1,00	0,95	0,55	0,95	0,55
	2xx	0,89	0,81	0,78	0,86	0,59	0,39	0,59	0,39
	3xx	0,88	0,63	0,79	0,85	0,38	0,19	0,38	0,19

Fig. 10: Ontology matching results table

In table 10 we present the results of a preliminary experiment conduced in the OAEI2005 dataset, in order to validate our implemented algorithm. We also implemented a Simulated Annealing approach, however we concluded the genetic-algorithm performs better when solving this problem.

IV. Conclusion

The paper considered the importance of addressing contextual information for intention understanding in modern robots. We proposed a possible methodology to address the problem of understanding the user intention, by means of context identification based not only on gaze tracking, but also on gestures analysis. The gaze tracking informs the robot about the user's focus of attention, if any. The gesture analysis enables the robot to receive orders or information from the user in a simple manner, or simply to direct the robot attention to somewhere or something, depending on the context. Both gesture and gaze analysis have shown interesting results, and as the corresponding methods do not require excessive computational power they are adequate for integration in a mobile robot.

We believe that these two mechanisms can provide important information for achieving the construction of a robot that can infer about user intention. This will be possible through a context-dependent interpretation of these two interaction cues. To support our approach we have presented some implementation consideration and results of the different components of the whole solution.

References

[1] A. N. Glass and K. J. Holyoak, *Cognition*, 2nd ed. New York: Random House, 1986.

[2] R. Pevtzow and R. L. Goldstone, "Categorization and the parsing of objects," in *16th Annual Conference of the Cognitive Science Society*, 1994.

[3] B. A. Mellers and A. D. J. Cooke, "The role of task and context in preference measurements," *Psychological Science*, vol. 1996.

[4] E. C. Ferstl, "Context effects in syntactic ambiguity resolution: the location of presuppositional phrase attachment." in *16th Annual Conference of the Cognitive Science Society*, 1994.

[5] T. Holtgraves, "Communication in context: effects of speaker status on the comprehension of indirect requests," *Journal of Experimental Psychology*, vol. 20, pp. 1205–1218.

[6] R. M. Turner, "Context-mediated behaviour for intelligent agents," *Int. J. Human-Computer Studies*, 1998.

[7] N. Moray and I. Rotenberg, "Fault management in process control: Eye movements and action," *Ergonomics*, vol. 32, no. 11, pp. 1319–1342, Nov. 1989.

[8] Baron-Cohen, *Mindblindness*. MIT press, 1995.

[9] P. H. S. Jun Su Ha, "Attentional-resource effectiveness measures in monitoring and detection tasks in nuclear power plants," *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol. 40, no. 5, pp. 993–1008, 2010.

[10] S. Asteriadis, P. Tzouveli, K. Karpouzis, and S. Kollias, "Estimation of behavioral user state based on eye gaze and head pose application in an e-learning environment," *Multimedia Tools and Applications*, vol. 41, no. 3, pp. 469–493, Feb. 2009.

[11] S. Mitra and T. Acharya, "Gesture recognition: A survey," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37, no. 3, pp. 311 –324, may 2007.

[12] P. Menezes, F. Lerasle, and J. Dias, "Towards human motion capture from a camera mounted on a mobile robot," *Image and Vision Computing*, vol. 29, no. 6, pp. 382–393, 2011.

[13] J. Y. Oh, C. W. Lee, and B. J. You, "Gesture recognition by attention control method for intelligent humanoid robot," in *Knowledge-Based Intelligent Information and Engineering Systems*, vol. Part I, 2005, pp. 1139–1145.

[14] M. Takahashi, K. Irie, K. Terabayashi, and K. Umeda, "Gesture recognition based on the detection of periodic motion." IEEE, 2010.

[15] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition (CVPR2001)*. Kauai, HI, USA: IEEE Computer Society, December 2001.

[16] H. Bay, T. Tuytelaars, and L. V. Goo, "Surf: Speeded up robust features," in *European Conference on Computer Vision (ECCV2006)*, Graz, Austria, May 2006.

[17] L. R. Rabiner, M. J. Cheng, A. E. Rosenberg, and C. A. McGonegal, "A comparative performance study of several pitch detection algorithms," *Transactions on acoustics, speech, and signal processing.*, vol. ASSP-24, no. 5, pp. 399–401, October 1976.

[18] *OpenNI User Guide*, OpenNI organization, 2010.

[19] *Prime SensorTM NITE 1.3 Algorithms notes*, PrimeSense Inc., 2010.

[20] D. Borrmann, J. Elseberg, K. Lingemann, and A. Nüchter, "The 3d hough transform for plane detection in point clouds - a review and a new accumulator design," *3D research, Springer*, vol. 2, no. 2, 2011.

[21] A. Tversky, "Features of similarity," *Psychological Review*, 1977.

[22] J. Wang, Z. Ding, and C. Jiang, "Gaom: Genetic algorithm based ontology matching," in *Proceedings of the 2006 IEEE Asia-Pacific Conference on Service Computing (APSCC06)*, 2006.