Towards A Human Robot Interaction Framework with Marker-less Augmented Reality and Visual SLAM

Eranda Lakshantha and Simon Egerton
Faculty of Information Technology, Monash University
Email: eranda.lakshantha@monash.edu; simon.egerton@monash.edu

***Abstract*—This paper presents a novel framework for Human Robot Interaction (HRI) using marker-less Augmented Reality (AR). Unlike marker-based AR, marker-less AR does not require the environment to be instrumented with special markers and so it works favorably for unknown/unprepared environments. Current state-of-the-art visual SLAM approaches like PTAMM (Parallel Tracking and Multiple Mapping) achieve this with constrained motion models within local co-ordinate systems. Our framework relaxes motion model constraints enabling a wider range of camera movements to be robustly tracked and extends PTAMM with a series of linear transformations. The linear transformations enable AR markers to be seamlessly placed and tracked within a global co-ordinate system of any size. This allows us to place markers globally and view them from any direction and perspective, even when returning to the markers from a different direction or perspective. We report on the model's performance and show how the model can be applied to help humans interact with robots. In this paper we look at how they can assist robot navigation tasks.**

***Index Terms*—augmented reality, human-robot interaction, robotics, SLAM.**

I. Introduction

Applications of Augmented Reality (a technology that can overlay virtual graphics on top of a real-time video feed), have great potential for bringing new avenues in human-robot interaction. We present a robust marker-less AR technique in which we could place an AR object at any point in space (as seen by the camera) and then associate a command to be fulfilled by the robot at that point. This is quite similar to the behavior of pointing somewhere in space and then giving an instruction like `move here' which could be often observed in human-human engagements.

While building such a HRI framework, marker-less AR is considered more suitable as opposed to marker-based AR since the latter approach requires the environment to be attached with physical markers which in turn imposes strict limitations to the robot's environment. On the other hand marker-less AR does not assume the target environment to be known in advance, thereby the robots could be readily operated in any environment. But most marker-less AR applications require a steady camera Field-Of-View (FOV) whereby the camera orientation does not vary more than a fixed amount i.e.180 degrees.

Extending the tracking robustness for unknown scenes was explored in [1] and [2]. However a continuous camera tracking is not always possible and, losing the AR poses as the camera changes viewpoints is common. Our prime concern is about robustly tracking marker-less AR under such circumstances and its application as a contemporary human-robot interface.

To put our question into perspective let's consider a practical scenario. Suppose a mobile robot mounted with a camera on top moves towards an AR object. It keeps on moving forward until it passes the AR object. Once it has travelled a slight distance away from the AR marker (i.e. AR object), it turns back and looks in the direction it travelled. At that instance the AR marker must be still visible. These AR markers persist in the environment until they are removed and so appear and behave as real world objects would do through the camera of the robot.

This notion of persistence enables many Human Robot Interaction tasks such as navigation, programming, multi-robot/multi-user collaboration and so on. In this paper we report on HRI navigation use as it is one of the most universal robotic operations. By connecting a series of persistent AR markers an AR navigation path can be overlaid on to the camera view and become a powerful interface in remote robot operations especially for people who have little to no expertise interacting with robots, for example navigating a domestic robot at home or navigating a service robot around the factory floors.

We develop our work by optimizing PTAMM [1] which is the current state-of-the-art. It is known to be the most suitable approach for tracking camera pose in previously unknown environments. We extend PTAMM to map in a global space of any size and call the extended model Parallel Tracking And Global Mapping (PTAGM). We show empirically that persistent AR markers are useful and robust. Later sections of this paper shows PTAGM's operation as a human-robot interface; thereby showing how to guide a robot towards a designated location in space using marker-less AR.

Manuscript received August 1, 2013; revised November 6, 2013.

doi: 10.12720/joace.2.3.250-255

The next section outlines the related literature whereas later sections describe in detail the methodology, results obtained and its applications.

II. Related Work

A solid marker-less AR technique always demands a real-time and an accurate computation of the camera pose [3]. Maintaining such scene augmentations have been well explored by [4], [5] and [6]. However, these studies look promising only if a predefined knowledge of the environment is available; thereby they lack support for unknown or unprepared environments. On the contrary, our focus is to robustly track marker-less AR under unknown environments (i.e. less constrained and better for robotics).

[7] presented the notion of maintaining a growing map as the robot explores new regions. They managed to build a map based on the texture of a floor which was distributed in a uniform manner. Despite their limitedness for special textures, it appeared as a possible replacement for [4], [5] and [6]. This implies the potential of a `mapping' approach, like Visual SLAM[1] in order to substantiate the tracking quality of marker-less AR for robotics.

[8] emerged as the first visual SLAM approach in the domain of mobile robotics. Their work, dubbed as MonoSLAM presents a probabilistic 3D map which is dynamically updated by a EKF (Extended Kalman Filter). The map is built according to a metric scale, and so it always requires the system to initialize by looking at an object of a known size. This is a weakness of MonoSLAM that makes it discourageable for robotics applications. However it was no longer a problem with the advent of PTAM (Parallel Tracking And Mapping) [2].

Instead of a metric scale, PTAM maintained a map, built according to an arbitrary scale. Along with its other advantages like, no prior assumption about the environment, ability to sustain a growing map, and the presence of an external coordinate frame, PTAM grounds a solid base for our notion of bringing up marker-less AR as a human-robot interface. However, the experimental results with PTAM are seen questionable for its ability to perform in a persistent manner. Constant jittering took place whenever the camera was moved far away from its original location. This happened as it lost its pose and we observed it struggling to re-localize itself with the best known pose. It totally lost the AR object at a point where the camera orientation spanned more than 180 degrees.

To some extent, PTAMM dealt with this inadequacy by generating multiple local coordinate frames associated with multiple maps in a one-to-one fashion. These coordinate frames are produced on a dominant plane formed by its own map points. It gave the user freedom to move the camera around and pick multiple locations in the environment where they act as placeholders for generating local coordinate frames.

The supplementary video file[2] shows a typical operation of PTAMM. However PTAMM's maps operate independently and so they do not uniformly track a single AR marker in a global fashion. Instead PTAMM displays the AR object associated with the current local map. Furthermore, the individual local maps are seen to be switching among each other while delivering an irritated impression for the user, which in turn makes it hard for positioning the camera's *global* location within the environment. This effectively lead us to extend PTAMM's ability to maintain a single global pose throughout multiple local coordinate frames, which in turn could act as a robust interface for human-robot operations.

III. Our Methodology in the Context of PTAMM

The method we followed can be summarized as below.

- Global camera pose is maintained by generating a set of multiple local coordinate systems.
- A series of linear transformations is used to combine these multiple local coordinate frames.

To place our method into focus, it is useful comparing our method with both PTAMM and PTAM. In our knowledge PTAMM's argument of multiple maps emerges as a complimentary solution to address the weaknesses of its predecessor PTAM. Authors of both PTAMM and PTAM believed that maintaining a global map would be a resource intensive and a time consuming operation. The reason is that the bundle adjustment times linearly amplify as the number of map points grows up; which ultimately caused the AR object to be unstable under different view directions, often losing its visibility when the view angle changes over a great extent (i.e. more than 180 degrees).

On the other hand, we operate a single map at a given time, and preserve a common global pose among subsequent local frames (local maps) by *linear transformation*. We take the camera's initial location as the origin of the global coordinate frame and express the camera pose with respect to this global-frame origin. The system described here (i.e. PTAGM) supports forming new coordinate frames as the camera enters unexplored regions within the environment. Consequently the camera switches into a new coordinate frame at the presence of a new region; and in such a manner the total set of coordinate frames could resemble the entire area of camera motion. While switching between multiple coordinate frames, PTAGM is capable of maintaining a seamless camera pose in relation to the global frame by taking linear transformations into account. Inter-coordinate transformations are calculated with the following mathematical formula.

$$x' = [R \mid t]x \quad (1)$$

x' and x are vector representations of an identical point in a global frame and a local coordinate frame respectively. R denotes the amount of rotation in matrix form whereas t signifies the amount of translation in vector form between

[1] Simultaneous Localization And Mapping based on computer vision

[2] http://youtu.be/Qo62V7pd_Kk

the two Cartesian spaces. $[R|t]$ is collectively known as the transformation matrix which transforms a homogeneous point (here it is x) from one coordinate frame to another. Based on this principle our algorithm sets out a persistent way for tracking marker-less AR under unknown scenes, while capturing the rotation and translation of consecutive local frames in relation to a fixed global frame.

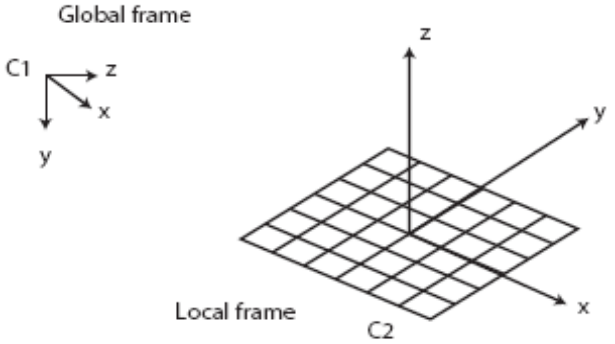

Figure 1. Global frame C1 and local frame C2. The camera initially lies within C1

Fig. 1 depicts the initial setup during our algorithm's operation. The system produces two coordinate frames during its initialization namely C1 and C2. The camera initially remains at the origin of C1 and so C1 is considered as the global frame. C2 on the other hand represents the local map (local frame) which in turn denotes the current region being operated. The rotation and the translation of C2 in relation to C1 is given by R_i and t_i and these values are stored inside the current map. As mentioned above, our algorithm permits to create more local coordinate frames, whenever the camera enters into new regions. However these local coordinate frames describe only the region of current camera operation, and computes only a local camera pose in the context of the currently operating local map. As a result, the camera pose changes frequently each time it switches into a new local coordinate frame - which is problematic for persistent tracking. Thereby we need a mechanism for maintaining a steady and a continuous camera pose while switching between different coordinate frames. In doing so, our system employs an inter-mediate coordinate frame for each local frame, whereby it acts as a bridge to connect with the global frame C1. The rotation and translation of such an inter-mediate frame (in relation to C1) is given by $R_i^{'}$ and $t_i^{'}$. Consider the example illustrated in Fig. 2.

Fig. 2 depicts the camera motion through coordinate frames C1 to C3 and the creation of a new local frame C4. C3 represents an inter-mediate coordinate frame which connects C4 with C1. Here $R_i^{'}$ and $t_i^{'}$ signifies the rotation and translation of C3 with respect to C1. It is important to realize that the camera now operates totally on a new region, identified by the local frame C4. Therefore the ultimate result - the camera's global pose could be obtained by finding the coordinate transformation [3] of C4 with respect to C1. In other words, the camera's position and orientation has to be expressed in terms of the global frame C1.

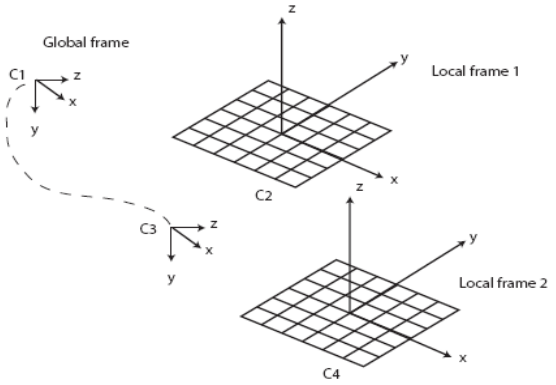

Figure. 2. Producing another local coordinate frame C4

In doing so, our method follows a two-step approach. First it finds the transformation of C4 with respect to C3. The goal of this step is to express the camera pose in terms of C3. Then it computes the transformation of C3 with respect to C1, hence delivering the camera pose in terms of C1. We hereby outline the equations occupied for these transformations in the order as they operate. Two variables are introduced namely R_{cur} and t_{cur} to indicate the local camera pose which is associated with the local map C4. Assuming the camera is positioned as in Fig. 2, transformation from C4 to C3 can be given by,

$$t_{C3} = R_i.t_{cur} + t_i \tag{2}$$

$$R_{C3} = R_{cur}^{-1}.R_i \tag{3}$$

where R_{C3} and t_{C3} are rotation and translation (pose) parameters of the camera with respect to C3. Once we have equations (2) and (3) in place transformation from C3 to C1 can be computed as,

$$t_{C1} = R_i^{'}.t_{C3} + t_i^{'} \tag{4}$$

$$R_{C1} = R_{C3}^{-1}.R_i^{'} \tag{5}$$

Finally the values of R_{C1} and t_{C1} constitute the camera's global pose. Our algorithm iteratively computes the equations (2), (3), (4) and (5) at frame rate which results in carrying out a series of linear transformations throughout multiple coordinate frames (maps). To visualize a better picture of our algorithm we exposed its steps in Fig. 3.

IV. Results & Experiments

Evaluations are performed within an indoor space using a standard web-cam. First, we testify PTAGM's tracking robustness for marker-less AR while taking PTAM as a benchmark, since PTAM and our system (i.e. PTAGM) shares the common trait of having a single global map.

[3] Expressing a coordinate system in terms of another one

However, we could not take PTAMM for these comparisons as it operates on a set of multiple local maps - instead of the notion of a single global map. Secondly, we investigate the application of our system as a human-robot interface - specially for robot navigation, in which we navigate a robot within a less constrained environment. Section A in the following describes the tracking robustness of PTAGM whereas section B demonstrates the use of PTAGM in HRI navigation.

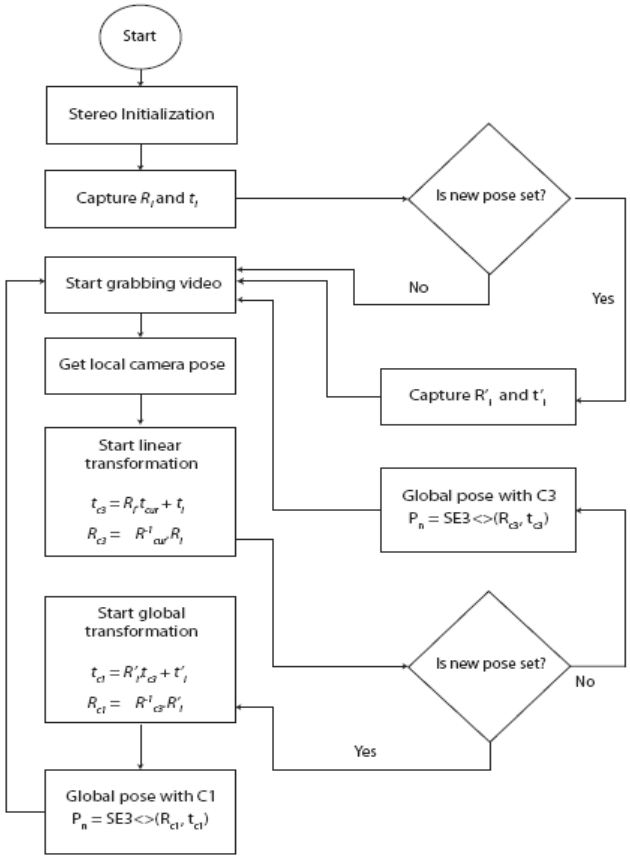

Figure 3. PTAGM algorithm

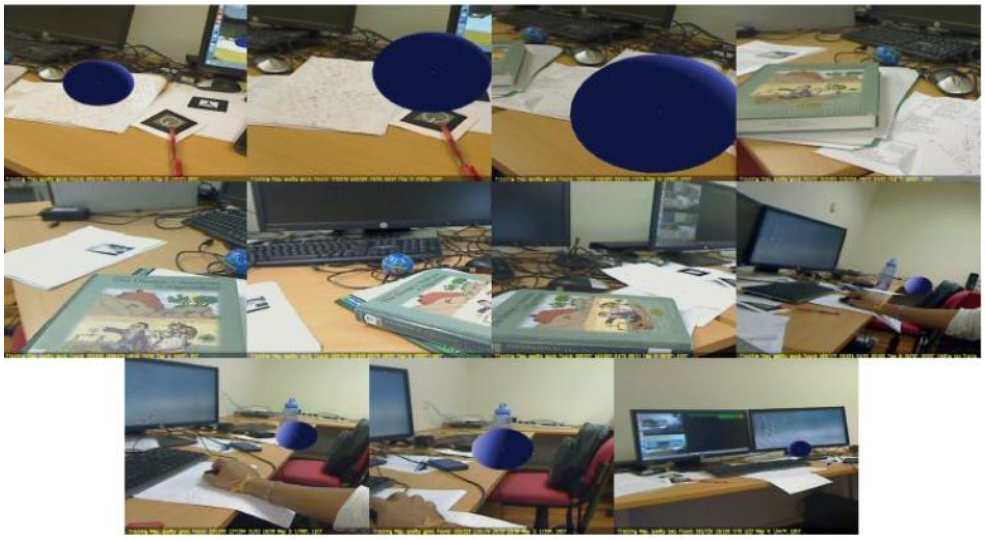

Figure 4. Persistent tracking of an AR object for wider camera perspectives

A. *Tracking Reliability for Wide Camera Motion*

Tracking reliability of both PTAM and PTAGM was put into test under similar conditions. After the system initialization, the camera was moved towards the AR object. We kept on moving the camera in a straight line until it went pass the AR object. Still a steady motion was taken forward and finally turned the camera around with an angle of 180 degrees. At this instance the camera was looking back at the path it travelled. The argument to

validate at this point is that the camera should still see the AR object persistently anchored at its original location without any jitter. The results are showcased for PTAGM in Fig. 4 where each individual image represents the steps mentioned above. The full demonstration can be observed in the accompanying video file[4]. It can be seen that PTAM lost the sphere like AR object at the end of the sequence whereas PTAGM displayed it persistently. Such a qualitative comparison during a live run happens to give notable results for PTAGM over PTAM.

Video file[5] for PTAM's operation further justifies this distinction. (Note that we have modified the PTAM's GUI slightly to bring both systems into equal grounds). PTAGM generated five maps during this experiment with each map holding 31, 26, 5, 82, and 94 keyframes respectively. On the other hand feature points are recorded as 2446, 1984, 861, 221, and 540. Therefore the largest map generated here is map 0 with 2446 feature points. Its bundle adjustment times are illustrated in Fig. 5. Note that the bundle adjustment time keeps on rising as the number of feature points grow up. The vertical dotted line indicates the point where the camera switched into a new local coordinate frame. Even though we used a hand-held camera for this test run, the same results can be imagined for a robot moving with a camera on its top.

B. Applying PTAGM for Robot Navigation

For this test scenario we employed a ground robot and a camera that outlooks the robot's target environment. Our marker-less AR interface (i.e. PTAGM) lies in between the human operator and the robot. In this set up, the AR interface enables human operator to point to a location in the scene displayed by the camera and then navigate the robot towards that location. In this case, the act of `pointing' to a location is accomplished by displaying an AR object at the location being pointed. The accompanying video[6] demonstrates how this functions.

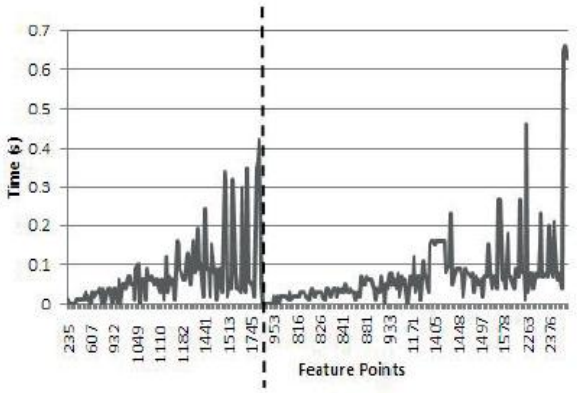

Figure 5. Bundle adjustment times for map 0

Such a `point and go' manner is a natural mode of communication for humans as it closely relates to expressions like `Move here' and `Go there', which is often observable in typical human-human interactions. Nevertheless the robot needs the distance and rotation required to reach the target location designated by the AR object. The global coordinate frame in our system assists in computing these parameters in terms of the coordinate frame's units. However the robot only accepts metric distances, thereby a correlation between the PTAGM's global frame and the metric scale is needed. This is achieved by sliding the robot 10cm at the system's outset. The correlation process operates only once since we maintain a global pose and the metric proportion remained unchanged throughout the entire application runtime. This demonstration is done with Eddie robot platform as illustrated in Fig. 6.

Figure. 6. Robot navigation with PTAGM. The top left-hand represents the robot's start location. Next shows the placement of the AR marker at a point in front. The bottom-most image depicts the robot's destination.

V. Limitations & Future Work

The system described here comes with some known issues and drawbacks. First it requires the user involvement for initializing a new map. This does not deliver a reasonable authentic value for HRI, especially for non-robotic experts. As a workaround we plan to incorporate two cameras with a fixed baseline, instead of having a single camera.

Despite using a single AR marker, the system must function with multiple AR objects. This is important since the user must be able to place several markers that act as a series of waypoints representing a navigation path. Nevertheless our system must not preclude HRI, only for navigation. Instead, it should support other areas of HRI such as multi-robot control, robot gaming, robot programming, learning from demonstration (LfD), etc. Further extensions to the current model are being

[4] http://youtu.be/cu7BIbyKMNc
[5] http://youtu.be/6AqdBQmyQJ0
[6] http://youtu.be/N0uQpxihSUo

underway for such a HRI task - multi robot control. A group of small robots could be effectively controlled by placing an AR object through PTAGM. For an example, a single AR marker can be shared among multiple robots and so a fleet of robots could be flocked at the location represented by the AR object. In such a manner, it reduces the operator's time to complete the task, rather than controlling each robot individually.

VI. Conclusion

Marker-less augmented reality has a great capacity to function as a novel human-robot interface. A 3D point in the robot's vicinity could be indicated with an AR object, while assigning a task to be fulfilled by the robot at that point. When combined with Visual SLAM, marker-less AR has the ability to render a virtual AR object in a readily available environment. Such environments are considered more natural for robot operations since highly controlled environments with fixed installations are not always guaranteed. But most marker-less AR applications find it challenging to confer a persistent tracking behavior in such environments as they fail to render the AR object firmly at its original location - specially when the camera view changes in a large angle. This hinders the usefulness of marker-less AR for HRI. In order to address this shortcoming, we present a system (i.e. PTAGM) based on PTAMM. Rather than being constrained by steady camera field-of-views, our system maintained a persistent global pose throughout wider and changing camera perspectives, while extending PTAMM with linear transformations. Having our marker-less AR interface in place, we show how it operates in the context of HRI by navigating a ground robot via virtual AR objects. Even though this study focuses robot navigation, expanding our approach to other areas of HRI, also seems plausible.

References

[1] R. Castle, G. Klein, and D. W. Murray, "Video-rate localization in multiple maps for wearable augmented reality," in *Proc. 12th IEEE International Symposium on Wearable Computers*, 2008, pp. 15–22.

[2] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2007, pp. 1–10.

[3] J. Herling and W. Broll, "Chapter 11 markerless tracking for augmented reality," *Handbook of Augmented Reality*, B. Furht, Ed. New York, NY: Springer New York, 2011, pp. 255–272.

[4] T. Lee and T. Höllerer, "Multithreaded hybrid feature tracking for markerless augmented reality," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 3, pp. 355–368, 2009.

[5] I. Gordon and D. Lowe, "What and where: 3D object recognition with accurate pose," *Toward Category-Level Object Recognition - Lecture Notes in Computer Science*, vol. 4170/2006, pp. 67–82, 2006.

[6] R. O. Castle, G. Klein, and D. W. Murray, "Combining monoSLAM with object recognition for scene augmentation using a wearable camera," *Image and Vision Computing*, vol. 28, no. 11, pp. 1548–1556, Nov 2010.

[7] G. Gini and A. Marchi, "Indoor robot navigation with single camera vision," in *Pattern Recognition in Information Systems*, 2002.

[8] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–67, June 2007.

Eranda Lakshantha received his B.Sc. degree in Information Technology from University of Moratuwa, Sri Lanka, in 2008. He worked as a software engineer since 2008 and is currently a Ph.D. Candidate in the Faculty of Information Technology at the Monash University. In 2012, he was awarded with a scholarship for PhD study at Monash University, Sunway Campus. His research interests are in the areas of Augmented Reality, Human-Robot Interaction, and computer vision with emphasis on extending AR assisted interfaces for HRI navigation and for other ecologies in HRI.

Simon Egerton received the PhD degree in Computer Science from the Department of Computer Science, University of Essex and conducts research on intelligent systems and bio-inspired robot control. He is a researcher who develops machine learning frameworks and techniques to advance the area of cognitive AI, applied to robotics, smart devices and their ecologies. He is the Deputy Head of School (Research) and leads the Intelligent Systems Research Laboratory. He is co-founder and director of the Creative Science Foundation, a charity dedicated to exploring the use of science fiction as a means to motivate and direct research into new technologies. He joined academia having spent a period of time in industry where he was responsible for designing, building and implementing embedded and real-time systems.