
Proceedings Paper:
Camilleri, D., Damianou, A., Jackson, H. et al. (2 more authors) (2016) iCub visual memory inspector: Visualising the iCub's thoughts. In: Biomimetic and Biohybrid Systems. 5th International Conference, Living Machines 2016, July 19-22, 2016, Edinburgh, UK. Lecture Notes in Computer Science, 9793 . Springer International Publishing , pp. 48-57. ISBN 9783319424163

https://doi.org/10.1007/978-3-319-42417-0_5

---

iCub Visual Memory Inspector: Visualising the iCub's Thoughts

Daniel Camilleri, Andreas Damianou, Harry Jackson,
Neil Lawrence, and Tony Prescott

Psychology Department, University of Sheffield,
Western Bank, Sheffield, United Kingdom
http://www.sheffield.ac.uk

Abstract. *This paper describes the integration of multiple sensory recognition models created by a Synthetic Autobiographical Memory into a structured system. This structured system provides high level control of the overall architecture and interfaces with an iCub simulator based in Unity which provides a virtual space for the display of recollected events.*

Keywords: Synthetic Autobiographical Memory, Unity simulator, Yarp, Deep Gaussian Process

1 Introduction

Human episodic and autobiographical (or event) memory can be considered as an attractor network operating in a latent variable space, whose dimensions encode salient characteristics of the physical and social world in a highly compressed fashion [1]. The operation of the perceptual systems that provide input to event memory can be analogised to a deep learning process that identifies psychologically meaningful latent variable descriptions [2]. Instantaneous memories then correspond to points in this latent variable space and episodic memories to trajectories through this space. Deep Gaussian Processes (DGP)[3] are probabilistic, non-parametric equivalents of neural networks and have many attractive properties as models of event memory; for example, the ability to discover highly compressed latent variable spaces, to form attractors that encode temporal sequences, and to act as generative models [4].

As part of the WYSIWYD FP7 project to develop social cognition for the iCub[5] humanoid robot, we are exploring the hypothesis that an architecture formed by suitably configured DGPs can provide an effective synthetic analogue to human autobiographical memory, a system that we call SAM. Work so far has focused on the development of models for separate sensory modalities that demonstrate useful qualities such as compression, including identification of psychologically-meaningful latent variables, pattern completion, pattern separation, and uncertainty quantification. The next phase focuses on the integration

of different sensory modalities using multiple sub-models which are cast within a single and coherent framework. However, with the use of multiple sensory modalities modelled together, one requires firstly a single point of entry for easy and intuitive communication with all models. This interface controls access to the separate sub models. Secondly, with the availability of multiple sub models, one also requires a method for visualising and associating all the recollected events in a virtual environment. This offers an important window into the recollection process of multiple models by visually displaying all recollected sensory modalities. We refer to this idea as the "visual memory inspector" (VMI) environment, to highlight the fact that we can *actively* (i.e. in a user-driven manner) interact with it and explore the memory space. The unique generative properties of the SAM model which we employ significantly facilitate the deployment of the VMI.

The development of this interface is also interesting since studies of human autobiographical memory indicate that whilst episodic memories are recovered via a loop through the hippocampal system the outputs of that system generate activity within primary sensory areas that appears to encode a sensory experience of the recollected event [6]. These patterns can then be picked up for processing elsewhere in the brain, for instance, by systems that plan future actions, or that reflect on the implications of remembered experiences. This activity also feeds through to hippocampus (as part of the loop) and may play a role in the reconstruction of further memories. Whilst the brain architecture underlying human autobiographical memory is poorly understood, our hope is that the development of an integrative architecture for autobiographic memory in a humanoid robot could provide clues for unravelling the role of different brain areas in human memory, and provide a top-down functional description of how such a system could operate.

The rest of this paper first provides an introduction to the operation of SAM in Section 2 together with a brief description of the models that have been trained so far based on DGP in Section 3. Section 4 subsequently outlines the need for a supervisory process to interface with multiple SAM Models. Section 5 describes the implementation of the Visual Memory Inspector crucial to the understanding of how the iCub is analysing the situation and finally Section 6 provides a description of the upcoming work on this project.

2 SAM Backend

In this section we first explain the SAM architecture used as a backend and in the subsections that follow, we outline the SAM-based sub-models developed in our work.

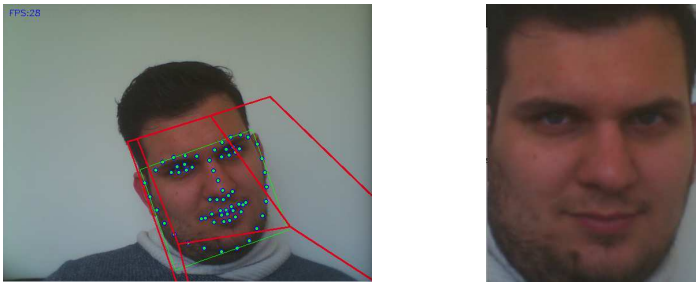
The SAM [2] system is a probabilistic framework which approximates *functional* requirements to Auto-biographical memory as have been identified in previous studies [1]. In detail, denote the N observed sensory data as D multi-dimensional vectors $\{\mathbf{y}_n\}_{n=1}^N$, i.e. $\mathbf{y}_n \in \Re^D$. Typically these vectors are noisy and high-dimensional, for example if the robotic agent is perceiving visual signals, each frame $\mathbf{y}_n$ will be a noisy image with D equal to the number of all the pixels composing it. In SAM, each $\mathbf{y}_n$ is modelled through $\mathbf{y}_n = f(\mathbf{x}_n) + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ is Gaussian noise. Here, $\mathbf{x}_n \in \Re^Q, Q \ll D$ is a low-dimensional vector, and $\{\mathbf{x}_n\}_{n=1}^N$ forms the (compressed) memory space (called a *latent* space) which is learned through the agents experience by Bayesian inference. Moreover, f is a Gaussian process mapping which maps latent points back to the original observation space. This is a *generative* mapping and plays a key role to the VMI. Furthermore, this mapping is anchored on a user-defined number of "anchor" points $\mathbf{U}$, meaning that any output of the function f will be a combination of elements from $\mathbf{U}$. [2] explains how the combination of anchor and latent points form the final memory space, where high-level analogies to neurons and synapses can be defined. In SAM, one can stack multiple latent spaces to form a hierarchical (deep) memory space. Notice that thanks to the Bayesian framework of SAM, we have access to the (approximate) posterior distribution $q(\mathbf{x}|\mathbf{y})$, meaning that once the model is trained we can readily obtain the reverse mapping of f when new sensory outputs $\mathbf{y}_*$ need to be considered.

We now proceed to outline the specific SAM-based sub-models which handle different types of sensory modalities. We will see later how these sub-models can be handled within a central framework which we call SAM Supervisor.

3 SAM Models To Date

3.1 Face Recognition Model

Face recognition with SAM has been demonstrated in previous work [7] which used a Viola-Jones face detector[8]. This method has been improved with the application of facial landmark identification and tracking through the use of a more robust face tracker called the Cambridge Face Tracker [9]. The output of this face tracker, depicted in Figure 1a provides an outline of the face together with a general direction of looking. This face outline is then extracted from the original image and processed into a rotationally invariant representation along the roll axis as shown in Figure 1b.



(a) (b)

Fig. 1: (a) Cambridge Face Tracker with light blue facial landmarks and red box representing orientation of the head. (b) Augmented output with roll rotational invariance

The image is subsequently resized, vectorised, labelled and then trained upon in the same manner as the previous work. Recollection is then carried out with the use of labels which returns a face extracted from the latent feature space that could either be a past observation or a fantasy face.

3.2 Tactile Model

The tactile model interfaces with the iCub's skin and collects the pressure reading over all texels of a specific body part, the arm, which are compiled into a single vector and trained upon to recognise four distinct types of touch which are:

1. Hard Touch
2. Soft Touch
3. Caress
4. Pinch

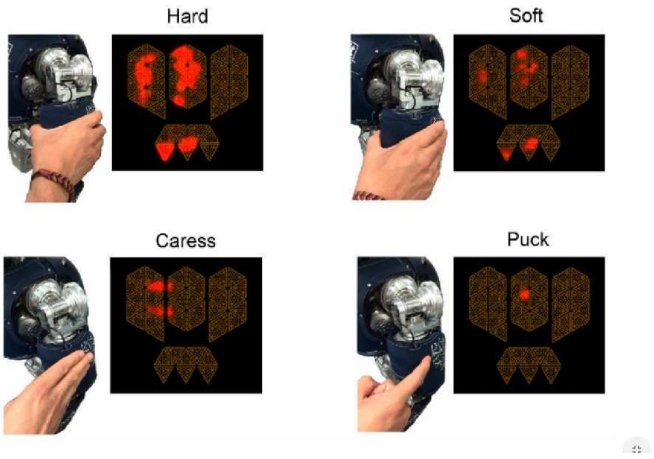


Fig. 2: Examples of the four types of touch on the iCub forearm classified by the Tactile Model

The recognition of these types of touch which can be seen in Figure 2 is particularly important in social situations. The recollection of a fantasy instance as the inverse process describes the pressure which is required for the re-enactment of the recollected touch.

3.3 Emotion from Speech Model

Another important indication which guides social interaction is the detection of emotional state especially from voice. As such there is currently work being carried out on the use of Mel-Frequency Ceptral Coefficients (MFCC) [10] paired with a Gaussian Mixture Model (GMM) to construct classification vectors for training with SAM.

In the first stage of feature extraction mel-frequency ceptral coefficients (MFCCs) are created for each frame of the utterance. These are a standard in speech processing and have shown great success in a number of tasks including speaker recognition and emotion recognition - as well as their ubiquitous use in speech recognition systems. MFCCs are approximations to the Fourier transform of the power spectrum of a frame of audio.

These MFCCs are then made into supervectors. Firstly, for each speaker, a GMM is trained on every feature from each of their utterances. These make up the speaker-specific Gaussian mixture models (GMMs). The feature vector is then generated using the method described in [11] which extracts the MFCC features for each utterance combined with the posterior probability of the mixture of Gaussians and this vector is used for training with SAM.

On the other hand, extraction of MFCCs from the raw waveform loses much of the original information necessary to recreating sound waves, such as intonation and other long-term features of speech. Thus the current state of the system does not allow the conversion of a recollection to sound.

However, this will be tackled in future work through the extraction of different features from sound which do not abstract the original audio signal as highly as MFCC features. One such feature that is being researched is the use of power spectra.

4 SAM Supervisor

The current challenge with multiple models of separate sensory modalities is the requirement to launch and interface with each individually. This hampers the development of more complex hierarchical models that link multiple modalities and this issue has led to the development of a streamlined system through the use of a supervisory process.

The aim of this supervisory process is, as mentioned in the introduction, to provide a single framework where all models that are developed with SAM can interact with the rest of the modules developed for WYSIWYD . As such the role of SAM Supervisor is fourfold:

1. Provide a single point of contact with all external modules accessing the models which greatly facilitates external interfacing. This exposes two valid commands for each model which are ***ask_modelName_label*** which returns the label given an instance of data or ***ask_modelName_instance*** which returns a fantasy memory instance given a label.

2. Initialise all models as subprocesses of the supervisor to ensure parallel operation and perform routine checks on the status of the loaded models.

3. Check that all models are up to date with respect to the available data (experiences) in the ABMSql database. This ensures that the iCub is current with respect to the conglomeration of its experiences to date.

4. Allow for the specification of model configurations that specifically describe which model configurations are to be loaded thus allowing custom memory layouts to be design and implemented easily.

Moreover, the presence of multiple models brings to light another challenge and that is understanding what is currently happening within the memory system which leads to the requirement of a Visual Memory Inspector whose implementation is detailed in the next section.

5 Visual Memory Inspector

The aim of the Visual Memory Inspector, as stated before, is to understand better what is currently occurring within the reasoning and recollection processes of the iCub in a visual manner. Thus this requires, first of all, a virtual world that behaves similar to the real world as it is understood by the iCub, a model of the iCub himself as the protagonist of this world as well as a means of communication with Yarp[12] for the transmission of information and motor commands.

As such this virtual world requires a platform with a physics engine to govern interaction between objects while also offering flexibility in interfacing with external libraries, cross-platform execution for both Windows and Linux and finally an easy way of developing applications within this platform.

The four candidates considered as a platform for the VMI were the Unity Game Engine[13], V-Rep[14], Webots[15] and Gazebo[16]. On one hand, Gazebo offers a versatile environment for the simulation of robots but requires the installation of ROS. V-Rep and Webots are also oriented towards the simulation of robots but are both difficult to interface to with Yarp and also have licensing restrictions and a small niche user base making development challenging.

Unity on the other hand satisfies all the requirements set for the VMI. It has an advanced GPU accelerated physics engine for simulating collisions and motion, allows multiplatform compilation because it derives from a .NET programming paradigm and furthermore facilitates the inclusion of external libraries in C# and/or JavaScript. Moreover Unity also has a vast user base which facilitates development and has no licensing requirements for the basic version which is versatile enough for the requirements of the project. Consequently after the choice of a development platform, the next step is to set up communication with Yarp from within Unity.

5.1 Unity-Yarp Integration

In order to integrate Yarp libraries within Unity, a common language is required to bridge the two platforms. Unity on one hand, can be developed using two languages, C# or JavaScript, of which JavaScript is easy to use but C# provides a higher level of control over the execution of code. On the other hand, Yarp is developed in C but it also provides language bindings for a variety of languages through the use of SWIG (Simplified Wrapper Interface Generator) [17] of which one of these languages is C#.

Thus with C# as the chosen language, the integration of Yarp with Unity is carried out by adding the Swig generated .dll (Windows) or .so (Linux) and .cs files that are generated through the compilation of Yarp to the Plugins folder of a Unity project and it is then imported as a library within the code.

With Unity capable of implementing Yarp ports the following crucial step for integration looked into the implementation of bottle and image conversion functions that allow decoding and encoding of Yarp information into a more Unity friendly format. Finally, since a call to yarp read is a blocking call, a threaded class was employed for the communication processes so that they can run in parallel to the visualisation. This results in higher frame rates and more time efficient processing of events. The next section describes the implementation of a virtual iCub as the protagonist of the memory inspector.

5.2 iCub simulation

The VMI is currently targeted towards the visualisation of memories and as such does not require a sensory interface within the virtual environment. Nonetheless, future work will look into expanding the scope of the VMI to a simulation space where future planning can also be virtually carried out.

This is why the current implementation of VMI pairs itself with iCub_SIM to allow motion control of the iCub within the VMI and also provides a stereo stream of the iCub's current point of view. This is accessible separately from the VMI interface which by itself provides a game like environment with the capability of walking around the 3D scene in the iCub's memory to change viewing angles.

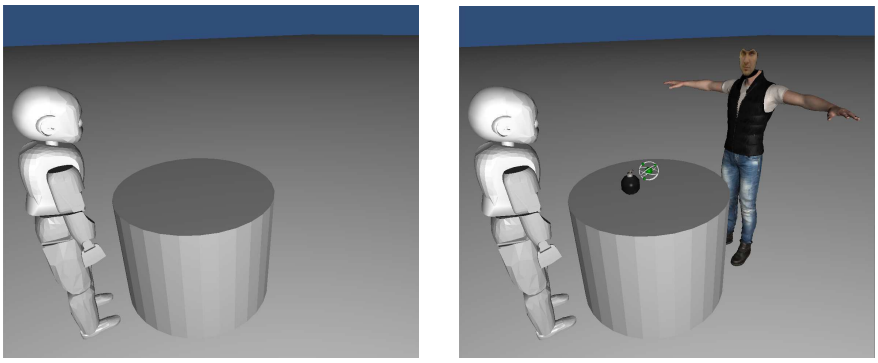
6 Future Work

6.1 Recollection Visualisation

The current state of the project allows the placement of the protagonist within a previously saved environment that could be obtained from a .obj model which includes Kinect generated 3D models. The VMI also has the functionality to dynamically load pre-existing 3D objects within the environment and assign a given label and position. An example of this can be seen in Figure 3 where the VMI has dynamically generated a person and two objects with specific locations within the environment.

The next major step in the implementation of the VMI is to integrate with the Language Reservoir developed by INSERM [18]. This module within WYSIWYD generates a Predicate-Action-Object-Recipient (PAOR) description of a previous memory retrieved through ABMSql. Each part of this concise description received by the VIM is then transmitted to SAM as an ***ask_modelName_instance*** request which returns a fantasy memory from the corresponding latent space.

Subsequently, after all constituents have been parsed and an instance received, the information is displayed as a scene within the VMI. A demonstrative example of such an interaction can be seen in Figure 4 where the face instance recovered from SAM for the label '*Daniel*' has been embodied within a generic body.



(a) (b)

Fig. 3: Demonstration of VMI dynamic object loading. (a) Depicts the initial state of the VMI which starts off with just the iCub and an environment (b) Depicts the state of the VMI with the dynamic addition of a person and two objects within the loaded environment

Fig. 4: Demonstration of the addition of a face recalled from SAM to the generic body of a dynamically instantiated person within the VMI

6.2 Virtual Sensing for Planning

Of the iCub's four senses, currently only sight is available within the VMI. Upcoming work will focus on the implementation of depth cameras, texels for touch and binaural microphones for sound which will allow planning and simulating the outcome of actions within the VMI.

7 Conclusion

This paper has briefly demonstrated the various applications that have been developed for SAM using different sensory modalities. Moreover, this paper has demonstrated the three challenges that arise with the concurrent use of multiple SAM models. These are the management of all models, the ease of interfacing and finally the challenge of visualising what is happening within these memory models in an interactive manner. As such we proposed the use of two modules: Sam Supervisor the Visual Memory Inspector (VMI) which provide a solution to this systems problem. Finally we lay out a plan for the continued development of these modules into an easily expandable software system upon which the development of a synthetic human autobiographical memory can be based.

References

1. Evans, M., Fox, C., Prescott, T.: Machines learning - towards a new synthetic autobiographical memory. In: Proceedings Living Machines. (2014)
2. Damianou, A., Boorman, L., Lawrence, N.: A top-down approach for a synthetic autobiographical memory system. In: Proceedings of the 4th International Conference on Biomimetic and Biohybrid Systems (Living Machines). (2015)
3. Damianou, A., Lawrence, N.: Deep Gaussian processes. In Carvalho, C., Ravikumar, P., eds.: Proceedings of the Sixteenth International Workshop on Artificial Intelligence and Statistics (AISTATS). AISTATS '13, JMLR W&CP 31 (2013) 207–215
4. Damianou, A.: Deep gaussian processes and variational propagation of uncertainty. PhD Thesis, University of Sheffield (2015)
5. IIT: iCub: an open source cognitive humanoid robotic platform. http://www.icub.org/ [Accessed 1-March-2016].
6. Gelbard-Sagiv, H., Mukamel, R., Harel, M., Malach, R., Fried, I.: Internally generated reactivation of single neurons in human hippocampus during free recall. Science **322**(5898) (2008) 96–101
7. Martinez-Hernandez, U., Boorman, L., Damianou, A., Prescott, T.: Cognitive architecture for robot perception and learning based on human-robot interaction
8. Viola, P., Jones, M.J.: Robust real-time face detection. International journal of computer vision **57**(2) (2004) 137–154
9. Baltrusaitis, T., Robinson, P., Morency, L.P.: Constrained local neural fields for robust facial landmark detection in the wild. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. (2013) 354–361
10. Zheng, F., Zhang, G., Song, Z.: Comparison of different implementations of mfcc. Journal of Computer Science and Technology **16**(6) (2001) 582–589
11. Loweimi, E., Doulaty, M., Barker, J., Hain, T.: Long-term statistical feature extraction from speech signal and its application in emotion recognition. In: Statistical Language and Speech Processing. Springer (2015) 173–184
12. YARP: Yet another robot platform. http://wiki.icub.org/yarpdoc/ [Accessed 10-February-2016].
13. Unity Technologies: Unity 5.2.2. https://unity3d.com/ [Accessed 1-March-2016].
14. Copelia Robotics: V-rep. http://www.coppeliarobotics.com/index.html [Accessed 1-March-2016].
15. Cyberbotics Ltd.: Webots. https://www.cyberbotics.com/overview [Accessed 1-March-2016].
16. Open Source Robotics Foundation: Gazebo. http://gazebosim.org/ [Accessed 1-March-2016].
17. Swig: Swig. http://www.swig.org/ [Accessed 1-March-2016].
18. Hinaut, X., Twiefel, J., Petit, M., Bron, F., Dominey, P., Wermter, S.: A recurrent neural network for multiple language acquisition: Starting with english and french