
Proceedings Paper:
Camilleri, D. and Prescott, T.J. orcid.org/0000-0003-4927-5390 (2018) Action recognition with unsynchronised multi-sensory data. In: 2017 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob). 2017 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob), 18-21 Sep 2017, Lisbon, Portugal. IEEE , pp. 53-59. ISBN 9781538637166

https://doi.org/10.1109/devlrn.2017.8329787

Action Recognition with Unsynchronised Multi-Sensory Data

Daniel Camilleri
Department of Psychology
University of Sheffield
Sheffield, UK
Email: d.camilleri@sheffield.ac.uk

Tony J. Prescott
Department of Psychology
University of Sheffield
Sheffield, UK

***Abstract*—Action recognition is a multi-faceted challenge that requires solving three principal challenges in its design. This paper discusses these principal challenges: Synchronisation, Segmentation and Uncertainty, together with their implications and possible solutions. We abstract the observations carried out for action recognition to generalise the challenges encountered in the classification of any time-dependant signal and finally propose the best performing approach as a general solution to this problem.**

I. Introduction

Action recognition is an important sensory problem that is crucial for the advancement of Human-Robot Interaction (HRI) or more generally Human-Machine Interaction (HMI). This is because, as a system, humans only have two outputs with which they can interact with the environment. These are either via the sensory modality of sound, like music or speech, or via actions that modify the physical environment. Action recognition thus allows the robot to understand better what is happening in its environment by being able to assign causal relationships between agent, action and outcome [1].

Action recognition is a heavily researched area that makes use of different sensory inputs such as vision [2]–[4] or depth [5]–[7]. Some choose to focus solely on actions which do not involve objects [8]–[10] or focus only on object related actions [11], [12]. Furthermore action recognition has a prerequisite: action segmentation, which is less heavily researched but just as important [13], [14].

In this paper we will be focusing on these challenges and others which arise from the use of an embodied system. We make use of action recognition as a target application while investigating the construction of this application in a general way that can be applied to other time-evolving signals such as EEG and EMG. We start in Section II by describing the characteristics of the input data stream and the challenges they impose. Then in Section III we discuss Synthetic Autobiographical Memory (SAM) [15], the modelling framework that we are using to build models for Action Recognition and how it allows us to better handle uncertainty and in Section IV the two methods of temporal segmentation that are tested on the iCub humanoid robot [16]. Finally, in Section V we present the results of the two temporal segmentation solutions and conclude in Section VI.

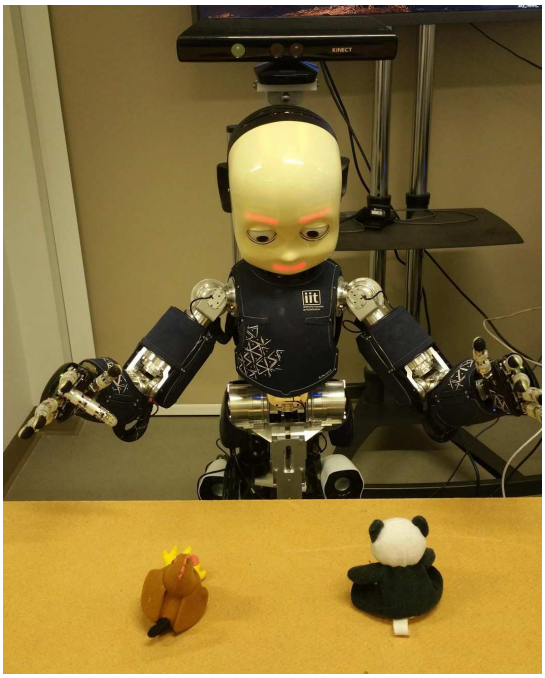


Fig. 1. iCub Setup for Action Recognition. Eyes focused on objects to track their location and kinect focused on the user to track the skeleton.

II. Challenges

A. Sensory Inputs

The first decision in the construction of an action recognition system starts with the decision of which data to use as an input for said system. Our implementation centres on performing action recognition with the iCub. Thus we have taken the approach to use both RGB data originating from the calibrated stereo cameras, the iCub's eyes, to track objects and depth data originating from a Kinect Version 1 which is mounted above the iCub as shown in Figure 1 to track the skeleton of the agent. This configuration was chosen not for the purpose of redudancy but for the purpose of seperately tracking objects and the skeleton of the interacting agent.

However, this configuration introduces intriguing challenges in the domain of multi-sensory integration which are heavily present in developmental robotics. The main challenges encountered are twofold. The first is that of combining different

sensory representations into a single representation for the purpose of training and modelling. The second challenge is that of synchronisation since two parallel data processing streams often have different throughputs which need to be rate-matched but also more significantly, different, sometimes variable latencies.

The latter is especially the case with the chosen configuration shown in Figure 1. The Kinect, on one side, has a high throughput of 30 - 40 frames per second (fps) and a relatively low absolute latency of around 100ms [17]. On the other hand, the objects are being tracked via the image processing pipeline of the iCub provided by IIT in the form of the Interactive Object Learning library (IOL) [18].

This pipeline uses Local Binary Patterns (LBP) to segment objects from the background and then employs the disparity map extracted in a separate module to assign a 3D position for the segmented object with respect to the eyes of the iCub. IOL also performs object recognition on the segmented images so that it tracks the names of the different objects within the scene. The throughput of IOL is between 20 and 30fps but has a high latency relative to the kinect which from observation varies between +600 to +800ms depending on the current computational load. Furthermore, since the IOL and the Kinect processing pipelines are running in separate modules that may be run on separate computers, one also incurs a non-determinant network latency at the receiving end.

This therefore presents a situation where the sensory streams cannot be accurately synchronised because the latency is variable. However, this configuration is analogous to the view of the brain consisting of multiple, parallel and asynchronous sensory processing streams [19] that are nonetheless combined together into a coherent representation of the environment. This challenge is referred to as the Synchronisation Challenge.

B. Action Segmentation

In order to carry out action recognition, one must first decide upon temporal segments of the incoming data stream that are representative of an action in order to process this data into a meaningful representation that can be used to classify this action. Thus, the automatic segmentation of actions from the incoming data stream is the second challenge that is addressed in our approach, the Segmentation Challenge, for which we have identified three possible solutions.

The first solution is to construct a model that is trained on the transitions between actions, in order to classify these transitions when they occur. This would signal a start and an end point for a temporal segment that can subsequently be used by the recognition model. This approach is very generic however its primary disadvantage is in the source of uncertainty. Any uncertainty present within the segmentation model would greatly increase the uncertainty of the recognition model.

This brings us to the second possible solution which takes the same approach of segmenting complete actions from the temporal stream but with the use of criteria such as motion in between scenes or the presence of contact between a hand and an object. This mitigates the problem of propagating uncertainty but with the presence of non-deterministic latency this approach suffers greatly because while one sensory stream can be indicating the absence of motion, the other sensory stream might be still catching up.

The final possible solution is in the use of temporal windows [20]. These are frames of fixed lenght that are extracted from the incoming data stream and processed by the recognition model. The use of this approach requires the recognition model to operate at a much higher frame rate than the other two solutions since there will be many more classifications per second required.

Furthermore, for the latter approach, due to the presence of multiple classifications per action one also requires a method of combining multiple classifications into a single decision. Despite this disadvantage, the latter approach increases the robustness of the classification and moreover draws a second analogy with the human brain, which as demonstrated by [21], recognises an action within 200ms of its start using partial trajectories.

C. Identifying Novel Inputs

The previous section describes the various methods that are available to carry out temporal segmentation of actions. However, detecting when an action starts and stops by employing the first two solutions does not guarantee that the temporal segment represents an action known by the model. This is also especially the case with the use of temporal windows where there is no guarantee that the temporal window chosen for classification is a valid action or even representative of an action because it could just as well be random motion.

This becomes an issue when a model is performing classification on this data because whatever the data represents, a model will still return one of the n labels that it was trained on. This is our third challenge, the Uncertainty Challenge.

One solution would be to return a probabilistic measure of certainty from the model together with the classification. One can subsequently apply a threshold and consider probabilities above the threshold as known and thus classifiable and those below the threshold as unknown and thus either ignored or stored for future learning.

This approach however can result in false negatives depending on how high the threshold is set. The model may be recognising an action correctly but returns an unknown result because of a low level of certainty. In some cases this is favourable but in other applications such as the early stages of a developmental learning approach, the high rate of false negatives would decrease the rate of learning.

In Section III we will demonstrate the various approaches that were tested on our model in order to address this challenge, the Uncertainty Challenge, from a developmental perspective.

III. Approach

Our approach is divided into five parts. First, we describe the dataset that is used throughout our modelling of action recognition. Then we describe how SAM allows us to model

this dataset followed by a description of the feature vectors used for the two temporal segmentation approaches that were tested for the Segmentation Challenge. Following this, we discuss how the different segmentation techniques also affect the Synchronisation Challenge and subsequently we discuss the solutions for the Certainty Challenge and their effect on classification performance.

A. Dataset

The dataset used is recorded from the setup shown in Figure 1 and consists of a total of 20,800 frames of data which are rate matched by upsampling the slow input stream to match the fast input stream. This data has been manually labelled on the basis of manually segmented actions and consists of 19 different labels that describe not only the action but also the name of the object that the action was carried on e.g. `push_object_car`. Of these 19 labels, 8 are chosen to be trained on which are:

1) `push_object_car`
2) `push_object_octopus`
3) `pull_object_car`
4) `pull_object_octopus`
5) `lift_object_car`
6) `lift_object_octopus`
7) `drop_object_car`
8) `drop_object_octopus`

These 8 labels account for 5401 data frames, 26% of the total in the form of 60 actions: 30 lift-drop pairs and 30 push-pull pairs. The rest of the actions are treated as unknown actions.

B. SAM

The starting point for the modelling approach we are employing was that of human episodic and autobiographical (or event) memory. This 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 [22]. The operation of the perceptual systems that provide input to event memory can then be analogised to learning processes that identify psychologically meaningful latent variable descriptions [23]. In this framework, instantaneous memories are seen as corresponding to points in the latent variable space and episodic memories to trajectories through this space. Seeding such a mechanism with appropriate clues will allow retrieval of a past episode, but the same system can also serve to fill-in and enrich the representation of the current situation, providing the potential for more informed action.

Gaussian Processes (GP) [24], [25] 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. The core element of our robot SAM system is therefore constituted by a set of GP latent variable models (GP-LVMs) that represent memories of multiple heterogeneous sensory modalities through a compressed latent feature space and a set of anchor points. Each SAM model knows how to combine these two elements to reconstruct past memory, or to generate fantasy memories (imagination). Chunking and pattern separation are also naturally manifested within this formulation. For instance, when a set of faces or actions is presented to the robot, memory formation naturally takes the form of clusters in the latent space, where separate clusters represent different faces/actions.

Our current implementation of robot SAM for the iCub humanoid robot is able to demonstrate effective memory formation and retrieval of human faces, actions, voices and emotions and is being progressed towards the challenge of representing sequences of agents acting on objects. Due to its generative nature, the system can also recreate memories leading to the possibility of imagining sequences such as an action by an agent that has not yet been observed [26]. By linking the sensory primitives of multi-modal memories (vision, sound, and touch), to verbal descriptions of episodes stored in the narrative processing parts of the system the SAM model could provide a way of grounding linguistic accounts of events in remembered experience [27].

Training a SAM model requires the training data to be of constant length, which is one of the drawbacks of the approach, together with an additional four parameters which are the number of inducing points that the model contains [28], the number of initialisation iterations, number of training iterations and the number of target dimensions for the output latent space called Q.

C. Segmentation Challenge

For the segmentation challenge, our approach makes a comparison of solutions 2 and 3 mentioned in Section II-B. For solution 2 we make use of two parameters, thresholded contact and the magnitude of object motion relative to its immediate past as the discerning variables to segment complete actions. Since actions are not always completed within a constant number of frames and since SAM requires a constant lenght feature vector, the segmented action is processed into the high level features shown in Table I as a description of the action for training. The described features are relative, thus during classification a list of hand-object combinations is created and a feature vector featuring all the features serialised into a single vector is constructed for each item in the list of possible combinations. This combinatorial approach was chosen because it would provide more information as to which arm has performed an action on which object within the scene.

On the other hand, in the case of solution 3 which makes use of temporal windows, there are also two parameters which define the segmentation. These are the window length and the window overlap which defines how many frames are skipped before the start of a new window as a percentage of the window length. The features extracted from these temporal windows in contrast with solution 2 are low-level physical features: position, velocity and acceleration of the hands and the objects. These are serialised and concatenated using the

Contact	This is defined by calculating a threshold distance between hand and object where values less than threshold correspond to 1 and values greater than threshold correspond to 0.
QTC Motion	The 3D version of Qualitative Trajectory Calculus(QTC) is defined in the paper by [29] based on the original work by [30] and is a method of describing relative movement between objects K(hand) and L(object) based on three outcomes: K approaching L (+), K stationary with respect to L (0) or K getting farther away from L(-).
QTC Orientation	QTC also has an orientation component which is defined by three angles in [29] that represent the orientation of K(hand) with respect to L(object) depending on the direction of movement of K. Each of these three angles are also expressed in terms of +, 0 and for each frame in the current action
Direction Vector	The direction vector represents the direction of the vector that connects the starting position of the data with the ending position of the current action.
Euclidian Distance	This is the distance between the starting position of the current action and its ending position
Relative Position Label	This classifies the position of the hand with respect to the object for each frame of the data as either: Top, Bottom, Front, Back, Left, Right
Relative Motion Label K	This classifies the movement of the current frame with respect to the previous frame for the hand according to the labels of Relative Position Label
Relative Motion Label L	This classifies the movement of the current frame with respect to the previous frame for the object according to the labels of Relative Position Label

TABLE I
DESCRIPTION OF THE HIGH LEVEL FEATURES THAT ARE EXTRACTED FOR SOLUTION 2

same combinatorial approach defined for solution 2. Due to the use of low-level features it was also decided to introduce two additional parameters. These are the size of a filtering window applied to the temporal window in order to smooth out the raw data and whether the features are absolute or relative to the start frame of the temporal window.

D. Synchronisation Challenge

The two approaches discussed in the previous section deal with the segmentation challenge using different methods. Now we discuss the effect of the different solutions with respect to the Synchronisation challenge. For solution 2, the synchronisation problem is somewhat bypassed by compressing the whole action into a set of high level features. However, since the features are relational in nature this worsens the effect of the synchronisation problem due to the computation of incorrect relational movement.

On the other hand, the effect of synchronisation on solution 3 is quite large due to the use of low level features which are heavily dependent on time. This is further aggravated with the use of a temporal window which only provides a fraction of an action. Thus the only control solution 3 has on the effect of synchronisation is the length of the temporal window. The greater the length of the temporal window, the more one can diminish the effects of synchronisation by learning to model the relative latency as part of the data.

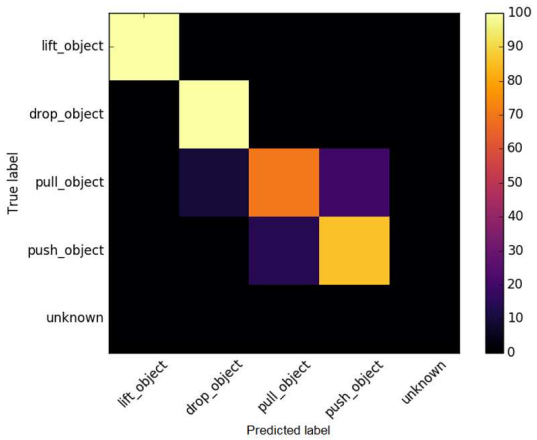
E. Certainty Challenge

As mentioned before, the actual actions that are trained only make up a quarter of the recorded data thus in a real world application, the trained model must be robust enough to ignore unknown actions when they occur and this is where the final challenge comes in. SAM's greatest advantage in this case is that since it is based on the use of Gaussian Processes, the model returns not only a mean value which corresponds with a classification label but also the Q-dimensional variance at that mean value of the Q-dimensional latent space.

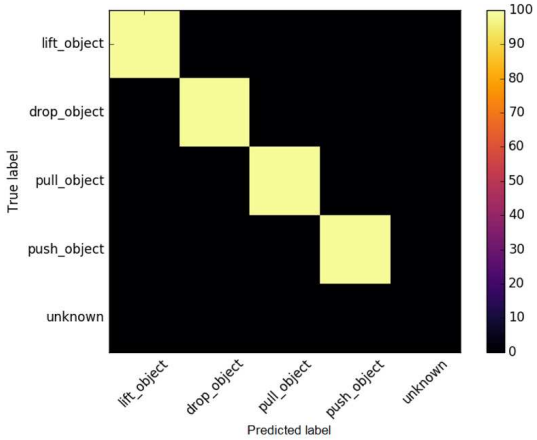
The variance is a good measure of certainty with a high variance indicating an unknown input while a low variance indicates a known input. However variance is not equivalent to probability because it does not have a bounded value but can vary in range even in between dimensions. Thus we proposed two methods that can transform the multi-dimensional variance into the probability that the action is known (P_{known}) and the probability that the action is unknown ($P_{unknown}$). Transforming the variances into two probabilities instead of one allows us to circumvent setting a fixed threshold on the probability but instead compare the two by taking the argmax as the winner.

1) Method 1: The first method assumes that the variances of all the known and unknown actions when combined together on a per dimension basis form a gaussian distribution with a mean and a variance. One can then compare the known and unknown distributions achieved per latent dimension and find in which dimension the distance between the two distributions is largest by calculating the bhattacharya distance [31]. Finally when the dimension with the largest distance is identified, one can take the variance of the classification result and from its value calculate the probability of that variance being a known variance or an unknown variance. This calculation is greatly simplified by finding the point at which the two gaussians intersect and using this value as the threshold value that decides between known and unknown.

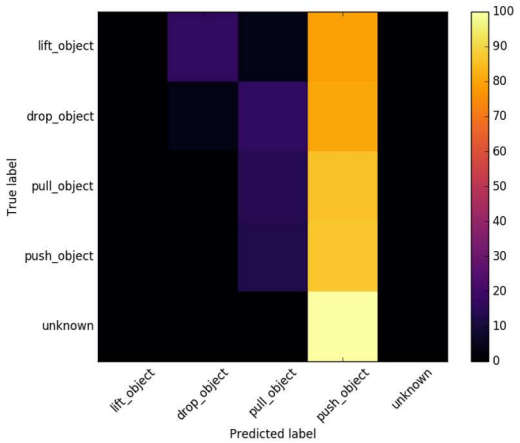
2) Method 2: Method 1 assumes that the distribution of the variances is a gaussian but this could be an incorrect assumption. So in Method 2 we take a different approach to deciding known and unknown. Instead of using a gaussian representation for the known variances and the unknown variance we instead calculate and store the per-dimension histogram of known variances and the per-dimension histogram of the unknown variances. Once normalised, these histograms provide a more accurate measure of the probability of the classification variance being known and unknown because they provide an experiential account of the distribution of known and unknown variances. This method thus takes the classification variance and calculates the probability of the variance being known or unknown on a per-dimension basis.



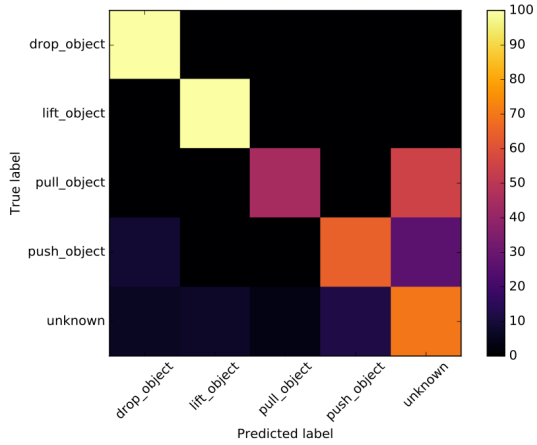
(a) Solution 2 Testing Known



(b) Solution 3 Testing Known



(c) Solution 2 Testing Unknown and Known



(d) Solution 3 Testing Unknown and Known

Fig. 2. Optimisation results for Solution 2 and Solution 3 for both Known only (2a, 2b) and mixed Known and Unknown conditions (2c, 2d)

This method introduces the number of bins in the histogram as an additional parameter.

Different means of combining these probabilities could also be considered. The first via a voting approach where the known and unknown of each dimension are compared and the argmax probability assigned a vote. The votes are then tallied across all dimensions and the highest tally is considered the winner. The second mechanism is taking the sum of all known probabilities and the sum of all unknown probabilities and taking the argmax of the results. This serves to marginalise the probability of $P_{known}|dimension$ into P_{known} and likewise for $P_{unknown}|dimension$ into $P_{unknown}$.

IV. Results

We present here the results obtained during our investigation of the Segmentation and the Synchronisation Challenge. We approached this by finding the optimal parameter configuration for both solutions 2 and 3 by feeding all the respective parameters into a Bayesian Optimisation implementation called GPyOpt [32]. Thus we find via Bayesian trial and error, the optimal configuration of model parameters and also the best combination of features to use, with the option of dropping features if deemed necessary to improve the overall classification. Both models are trained with 200 optimisation iterations in a search space of just over 3 million possible parameter combinations for solution 2 and 2 million for solution 3.

The results shown in Figures 2a and 2b are the results obtained for classification on the manually segmented actions in the dataset and they demonstrate that the model accuracy is very close to that obtained by solution 3, both of which are considerably accurate with low false positives and slightly highe false negatives. This shows that the negative effect of synchronisation has been indeed modelled by the GP in both cases. However for solution 2, when segmenting actions via the use of contact and motion thresholds which are also optimised, we obtain the results shown in Figures 2c and 2d. The segmentation approach of solution 2 thus fails completely when action segmentation is not robust enough in detecting proper action boundaries.

Solution 3 is therefore the best performing solution achieveing an overall accuracy rate of 75%

V. Conclusion

In conclusion we present here the three main challenges faced by any system processing a time varying signal for classification. The challenges of synchronisation, segmentation and uncertainty quantification. We demonstrate the various approaches one can take in overcoming these challenges and finally outline what we consider to be the most general and well performing solution to these problems. Future work will investigate the methods proposed for the Certainty challenge as well as a more thorough investigation into the role of window length with respect to the Synchronisation challenge. Furthermore we also plan to carry out a comparison with standard action recognition datasets to assess the performance of the best approach with respect to the state of the art.

Acknowledgment

Funding from EU Framework project WYSIWYD (FP7-ICT-2013-10) and from the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement no. 720270 (HBP SGA1).

References

[1] E. B. Bonawitz, D. Ferranti, R. Saxe, A. Gopnik, A. N. Meltzoff, J. Woodward, and L. E. Schulz, "Just do it? investigating the gap between prediction and action in toddlers causal inferences," *Cognition*, vol. 115, no. 1, pp. 104–117, 2010.

[2] M. Ahmad and S.-W. Lee, "Hmm-based human action recognition using multiview image sequences," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 1. IEEE, 2006, pp. 263–266.

[3] A. Bobick and J. Davis, "Real-time recognition of activity using temporal templates," in *Applications of Computer Vision, 1996. WACV'96., Proceedings 3rd IEEE Workshop on*. IEEE, 1996, pp. 39–42.

[4] R. Bodor, B. Jackson, O. Masoud, and N. Papanikolopoulos, "Image-based reconstruction for view-independent human motion recognition," in *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 2. IEEE, 2003, pp. 1548–1553.

[5] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3d points," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*. IEEE, 2010, pp. 9–14.

[6] X. Yang and Y. L. Tian, "Eigenjoints-based action recognition using naive-bayes-nearest-neighbor," in *Computer vision and pattern recognition workshops (CVPRW), 2012 IEEE computer society conference on*. IEEE, 2012, pp. 14–19.

[7] L. Xia, C.-C. Chen, and J. Aggarwal, "View invariant human action recognition using histograms of 3d joints," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*. IEEE, 2012, pp. 20–27.

[8] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local svm approach," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 3. IEEE, 2004, pp. 32–36.

[9] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2. IEEE, 2005, pp. 1395–1402.

[10] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action mach a spatio-temporal maximum average correlation height filter for action recognition," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[11] A. Gupta, A. Kembhavi, and L. S. Davis, "Observing human-object interactions: Using spatial and functional compatibility for recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 10, pp. 1775–1789, 2009.

[12] D. J. Moore, I. A. Essa, and M. H. Hayes, "Exploiting human actions and object context for recognition tasks," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 1. IEEE, 1999, pp. 80–86.

[13] M. Brand and V. Kettnaker, "Discovery and segmentation of activities in video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 844–851, 2000.

[14] F. Lv and R. Nevatia, "Recognition and segmentation of 3-d human action using hmm and multi-class adaboost," in *European conference on computer vision*. Springer, 2006, pp. 359–372.

[15] A. Damianou, L. Boorman, and N. Lawrence, "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.

[16] IIT. iCub an open source cognitive humanoid robotic platform. [Accessed 1-March-2017]. [Online]. Available: http://www.icub.org

[17] M. A. Livingston, J. Sebastian, Z. Ai, and J. W. Decker, "Performance measurements for the microsoft kinect skeleton," in *Virtual Reality Short Papers and Posters (VRW), 2012 IEEE*. IEEE, 2012, pp. 119–120.

[18] IIT. (2013) IOL interactive object learning. [Accessed 1-March-2017]. [Online]. Available: https://github.com/robotology/iol

[19] S. Zeki, "A massively asynchronous, parallel brain," *Phil. Trans. R. Soc. B*, vol. 370, no. 1668, p. 20140174, 2015.

[20] H. Jhuang, T. Serre, L. Wolf, and T. Poggio, "A biologically inspired system for action recognition," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. Ieee, 2007, pp. 1–8.

[21] L. Isik, A. Tacchetti, and T. Poggio, "Fast, invariant representation for human action in the visual system," *arXiv preprint arXiv:1601.01358*, 2016.

[22] M. H. Evans, C. W. Fox, and T. J. Prescott, "Machines learning-towards a new synthetic autobiographical memory," in *Conference on Biomimetic and Biohybrid Systems*. Springer, 2014, pp. 84–96.

[23] A. C. Damianou, M. K. Titsias, and N. D. Lawrence, "Variational inference for latent variables and uncertain inputs in gaussian processes," *Journal of Machine Learning Research (JMLR)*, vol. 2, 2015.

[24] N. Lawrence, "Probabilistic non-linear principal component analysis with gaussian process latent variable models," *Journal of Machine Learning Research*, vol. 6, no. Nov, pp. 1783–1816, 2005.

[25] A. C. Damianou and N. D. Lawrence, "Deep gaussian processes." in *AISTATS*, 2013, pp. 207–215.

[26] D. Camilleri, A. Damianou, H. Jackson, N. Lawrence, and T. Prescott, "icub visual memory inspector: Visualising the icubs thoughts," in *Conference on Biomimetic and Biohybrid Systems*. Springer, 2016, pp. 48–57.

[27] A. Damianou, C. H. Ek, L. Boorman, N. D. Lawrence, and T. J. Prescott, "A top-down approach for a synthetic autobiographical memory system," in *Conference on Biomimetic and Biohybrid Systems*. Springer, 2015, pp. 280–292.

[28] M. K. Titsias, "Variational learning of inducing variables in sparse gaussian processes." in *AISTATS*, vol. 5, 2009, pp. 567–574.

[29] N. Mavridis, N. Bellotto, K. Iliopoulos, and N. Van de Weghe, "Qtc 3d: Extending the qualitative trajectory calculus to three dimensions," *Information Sciences*, vol. 322, pp. 20–30, 2015.

[30] M. Hanheide, A. Peters, and N. Bellotto, "Analysis of human-robot spatial behaviour applying a qualitative trajectory calculus," in *RO-MAN, 2012 IEEE*. IEEE, 2012, pp. 689–694.

[31] G. Xuan, X. Zhu, P. Chai, Z. Zhang, Y. Q. Shi, and D. Fu, "Feature selection based on the bhattacharyya distance," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 4. IEEE, 2006, pp. 957–957.

[32] T. G. authors, "GPyOpt: A bayesian optimization framework in python," http://github.com/SheffieldML/GPyOpt, 2016.