Proceedings of the 2011 IEEE
International Conference on Robotics and Biomimetics
December 7-11, 2011, Phuket, Thailand

Learning Dynamical Representations of Tools for Tool-Use Recognition

Yan Wu, and Yiannis Demiris

Abstract— **We consider the problem of representing and recognising tools, a subset of objects that have special functionality and action patterns. Our proposed framework is based on the biological evidence of hierarchical representation of tools in the region of the human cortex that generates action semantics. It addresses the shortfalls of traditional learning models of object representation applied on tools. To showcase its merits, this framework is implemented as a hybrid model between the Hierarchical Attentive Multiple Models for Execution and Recognition of Actions Architecture (HAMMER) and Hidden Markov Model (HMM) to recognise and describe tools as dynamic patterns at symbolic level. The implemented model is tested and validated on two sets of experiments of 50 human demonstrations each on using 5 different tools. In the experiment with precise and accurate input data, the cross-validation statistics suggest very robust identification of the learned tools. In the experiment with unstructured environment, all errors can be explained systematically.**

I. INTRODUCTION

Traditionally, recognition of objects has been an important topic of research in computer vision, statistical image processing and robotics. Although humans can recognise a huge range of objects in an image without much effort, this task remains a challenge in machine intelligence in general. In the past decades, many approaches have been proposed to tackle this challenge [1], [2]. Most works so far focus on static matching of an object from a given image or video.

Tools, an invention of humanity since two and a half million years ago, are devices created with one or more specific functionalities to ease the process of producing or achieving a task. There are two main properties of tools making it a unique challenge for recognition:

1) Tools have a great variation of shapes. For example, an Alaskan Ulu knife has distinctly different appearance from a traditional chopping knife; a precision screwdriver looks more like a pen than a normal screwdriver, but the functionality and the way to operate them are the same in both cases. To tackle this problem in image recognition, a large database of tool images has to be supplied to a given learning model with scalable recognition algorithm, such as [3]. However, this might give rise to a misclassification in the problem define below.
2) Tools may have more than one functionality. For example, a claw hammer is a fusion of two tools, a claw and a hammer; a Victorinox tool comprises more than 6 tools. This makes object labelling of such tools a challenging task in any static matching framework.

Yan Wu and Yiannis Demiris are with the Department of Electrical & Electronic Engineering, Imperial College London, United Kingdom {yan.wu08, y.demiris}@imperial.ac.uk

Misclassification of tools can lead to dangerous or socially inapt moves by the robot in a human-robot interaction set-up. On the other hand, in everyday life, humans seem to be able to identify a familiar tool without being presented with much visual information. For example, a person performing a sawing action at far sight can be easily believed to have a saw at hand. This is supported by recent studies in neuroscience which suggest that identification of tools is done in the same brain region for action word generation ([4], [5], [6]). Thus, it is reasonable to assume that representation of tools in human cortex is closely related to the actions such tools can generate, i.e. having a goal-directed dynamic description. The only related work we have seen is done by Nishide et al [7] on tool-body assimilation for reaching actions.

In this paper, we present a biologically-inspired hierarchical learning framework for dynamic representation of tools. It can serve as a building block in tool-use learning for recognising tools in action, even when the environment is cluttered and the tools are lack of discriminative features. In the following sections, we discuss some background work towards building a dynamical representation before presenting our framework, the derived model and the detailed implementations. We then describe and discuss two experiments to verify the implemented model in identifying tools in-action demonstrated by human subjects.

II. RELATED WORK

Although representation of general objects in computer and robotic systems has been extensively studied ([2], [8], [9]) and shows promising performance in real-time[10] for recognising objects in a real-life environment, such algorithms often match static invariant features in an image with given objects. These kinds of invariant features might not work properly in categorising tools which might not have a fixed shape, colour or other static features within the same tool category. Furthermore, activations in our brain when a tool stimulus is presented suggest that tools are not only represented as objects in our brain but also associated with the actions they can perform ([5], [6]). In [11], a Bayesian algorithm is proposed to recognise objects based on human interaction with these relatively static objects in the scene making it unsuitable for representation of tools.

Grafton et al argue that there exists a hierarchical topology in the human brain in goal-directed action generation [4]. From the experimental observations, this hierarchy consists of 3 levels. At the top of the hierarchy, the desired *outcome* defines the consequence of the action. This is followed by the *goal-object* being manipulated in order to achieve the outcome. The *kinematics* of the system is at the lowest level.

978-1-4577-2137-3/11/$26.00 © 2011 IEEE

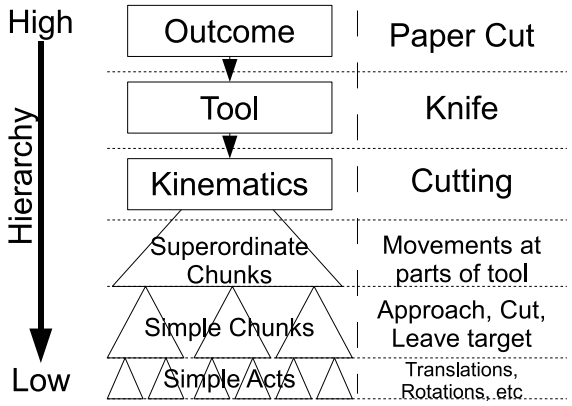


Fig. 1: **The schematics of the DTR framework.** The diagram on the left shows the framework from the highest level in the hierarchy to the lowest one. The diagram on the right is an example for illustration.

In [12], Koechlin et al propose that the Broca's Area of the human cortex organises kinematic actions into a further hierarchy of *superordinate chunks*, *simple chunks* and *simple acts*. We thus hypothesise that affordance, the dynamical property of tools can be used to classify them effectively. Although there has been much research effort in learning object affordance ([13], [14]), very limited work has touched specifically on tool affordance [15]. However, this work focuses on learning the tool affordance by motor babbling but not use this dynamical property for tool recognition.

In this paper, our investigation is inspired by and based on the biological evidences presented in [4] and [12] since representation of tools are closely associated with action description in the human brain. The common keyword of these two pieces of work is hierarchical framework. Among studies of hierarchical learning frameworks on action perception and execution for robotic systems ([16], [17], [18], [19]), the HAMMER architecture proposed by Demiris et al [18] emerges as a promising model. This primitive-based generative model does not require extensive training for generalisation which might not be practical to a physical robotic system. It is also one of the very few models that have been implemented with a more complex hierarchy based on biological evidence [20].

In [7], Nishide et al argue for the use of active exploration for learning. Although this approach shows good generative capability with unknown tools for 2D reaching action, the model does not scale well with multiple affordances and dimension of action space. On the other hand, programming-by-demonstration (PbD) [21] enables systems to acquire the generative capability in learning, recognising and applying new skills. It also provides an implicit learning environment to programme the robot and reduces the size of associated search-space. We therefore believe that PbD can serve as a basis for primitive extraction in our dynamical tool representation framework for the system to build up a more complete representation of tools over time.

To enable a system to generate new primitives based on observations, we require a model to describe the observations by a subset of basic or learnt primitives. The Hidden Markov Model (HMM), a dynamic Bayesian model capable of modelling observations in temporal sequences, has many successful applications in speech recognition, gesture recognition and robotic trajectory learning. In an attempt to classify object based on its affordance [22], Gupta et al propose an HMM-based model similar to gesture recognition approach. However, this system assumes that the object afforded on has similar movement to that of the human hand which is not generally true for tools. In our approach, we make use of multiple HMMs as a means to generate new dynamic description of parts of a tool and integrate them with the HAMMER architecture.

III. THE DYNAMICAL TOOL REPRESENTATION FRAMEWORK

A. The general framework

Fig. 1 shows the Dynamical Tool Representation (DTR) Framework. To learn and represent a tool in this framework, we first present a labelled tool in action to extract and label its kinematics and represent as a set of superordinate chunks which can be further broken down into a combination of simple chunks and simple acts. The outcome of the action on the target is also evaluated and labelled to connect the 3 parts in the hierarchy together, e.g. "Paper Cut"-"Knife"-"Cutting". To identify a tool, we evaluate the outcome and the kinematics of a demonstration. If there exists a one-to-one relationship between them via a tool recorded in the model, we say that the tool is successfully identified.

B. Derived Model from the Framework

In this work, we showcase one working model of the framework to extract the kinematics of the action for a given tool represented dynamically, by assuming the outcome of the action of a tool has a one-to-one relationship with the action kinematics via a tool. This assumption does not prevent the notion of multiple functionalities of a tool. Thus, a tool is deemed identified once the kinematics is recognised.

In a given demonstration, we define the coordinates of the centre of areas of interest on the tool as superordinate chunk $\mathbf{SC}_i^t, i \in (1 \dots N), t \in (1 \dots T)$, N is the number of areas, T is the total time-steps of the demonstration. We also denote the centre of the target a tool effects on as $\mathbf{D}^t$. For each area i, we construct a model to represent the kinematics of this superordinate chunk action for this area.

To segment each time-series of the superordinate chunks $\mathbf{SC}_i^t$, we use the HAMMER architecture of inverse-forward model pairs to denote the characteristics of all the simple chunks. These model pairs are arranged in parallel to direct attention of the system to learn a particular simple chunk. HAMMER requires a confidence measure to evaluate each model pair, and outputs the most probable symbol at each time-step. For example, in the case of "Approach", "Action" and "Departure" segmentation, the confidence measure is based on the change of mean and variance of the distance between the tool and the target.

At the lowest level of primitives, we use the HAMMER architecture as parallel simple act detectors at each time-step, assigning to each $\mathbf{SC}_i^t$ point the symbol of the most probable inverse-forward model pair. For instance, these can be based on the velocity vector, $\mathbf{v}_i^t = \mathbf{SC}_i^t - \mathbf{SC}_i^{t-1}$, such as stationary, translation and rotation about one or more axes. As compared to many other models, HAMMER at this level can act as a fast conditional-reflex mechanism allowing fast evaluation of multiple models with easy reference to previous experience.

Unless each simple chunk only consists of one simple act, we need a model to describe the dynamic transition from one simple act to another within a simple chunk. Since our simple acts are represented at symbolic level, this makes it convenient and practical to use HMM to learn the dynamics of simple chunks.

C. The Implementation

Fig. 2 shows the implementation of the model. In this implementation, we assume that a tool is a rigid extended degree of freedom (DoF) of a human operator, thus limit the superordinate chunks to $N = 2$, i.e. the handle and the effector areas of the tool. We implement a fixed topology for the kinematics hierarchy and a fixed number of HAMMER inverse-forward primitives. At the Superordinate Chunks level, the HAMMER segmentation model pairs are the three example models discussed in Section III-B. Transition from one superordinate chunk to another is left-to-right only, e.g. system at "Action" stage is not allowed to go back to "Approach" stage.

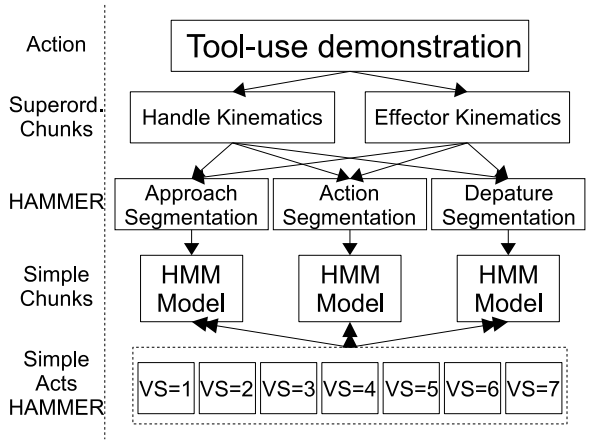


Fig. 2: **The implemented model of the framework.** $VS \in (1 \dots 7)$ is the velocity symbol generated by the lowest level of HAMMER models.

At simple chunk level, we use discrete HMMs with emissions symbols VS, each corresponds to a simple act and is supplied by the HAMMER architecture one level below. The use of discrete representation reduces the noise in detection, eliminating the need of additional filtering computation. We define a strategy to choose the desire number of states, $\min(Ns)$, for each HMM, where Ns is defined as:

$$\frac{\sum_{j=1}^{Ns} max_j(ct(1,\{VS^t\})ct(2,\{VS^t\}) \dots ct(Es,\{VS^t\})}{\sum_{j=1}^{Es} ct(j,\{VS^t\})} \geqslant \lambda \quad (1)$$

where $\lambda (0 \leqslant \lambda \leqslant 1)$ is a predefined threshold, VS^t is the sequence of emissions of in a simple chunk and Es is the last emission symbol. The operator $ct(k,\{VS^t\})$ returns the number of symbols that correspond to symbol k and max_j(No. array) returns the j^{th} largest number in the array.

Since all superordinate chunks are areas of interests on a tool, i.e. they are in a fixed geometric relationship, assuming tools are rigid bodies, the number DoFs for each chunk in a 3-dimensional space can be reduced to 3. In a goal-directed scenario, assuming there exists a target plane where the target lies on, these DoFs are $\hat{\mathbf{n}}$ normal to the target plane, $\hat{\mathbf{t}}$ tangential to the target in the plane and $\hat{\mathbf{r}}$ radial to the target in the plane, such that $\mathbf{n}$, $\mathbf{t}$ and $\mathbf{r}$ are orthogonal to each other. At the lowest level of simple acts, the maximum number of simple act detectors is 7, stationary and 2 directions in each DoF. Thus, we define 7 inverse-forward model pairs corresponding to stationary, approach target tangentially, leave target tangentially, approach target normally, leave target normally, rotate clockwise about the target and rotate anticlockwise about the target. The confidence measure is the probability of the velocity vector $\mathbf{v}_i^t$ moving in the direction.

To generate the probability for each primitive model at a given time-step, we propose a confidence-sharing mechanism within HAMMER, i.e. the past confidence of all models are published. The stationary model returns only binary confidence values (0 or 1) and inhibits all other models when stationary vector is detected. Excluding the stationary model, the probability p_t^m of a model m is updated by:

$$p_t^m = \frac{vel_{t-1}^m \times p_{t-1}^m + vel_t^m \times c_t^m}{\sum_i (vel_{t-1}^i \times p_{t-1}^i + vel_t^i \times c_t^i)} \quad (2)$$

where t is the current time-step, vel is the component of the velocity vector $\mathbf{v}_t$ in the direction of the model and c is the raw confidence value of vel.

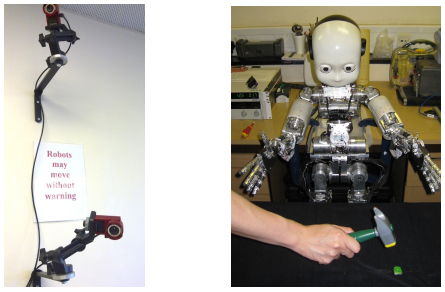
IV. EXPERIMENT

This implementation of the DTR framework was implemented and evaluated on two systems. In Experiment A, for statistical verification of the model, we used the Naturalpoint OptiTrack motion capture system to obtain a precise set of kinematics. This system consists of 8 OptiTrack FLEX-V100R2 cameras (Fig. 3a) with frame rate of 100Hz. Throughout the experiments, the static localisation error is around 1 mm. The same set of experiment was then conducted in Experiment B, but on the open-source iCub robot (Fig. 3b), a humanoid robot developed by the RobotCub Consortium[1], to test the performance in real life. The on-board stereo cameras was set at 10Hz frame-rate and 320×240 pixel resolution.

A. Experimental Scenario

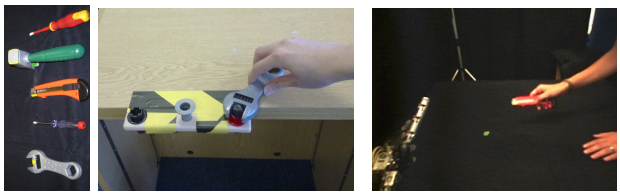
Each experimental subject has to demonstrate how to use 5 different tools in front of the iCub. The range of tools are shown in Fig. 4a. The demonstrator is required to choose only either one of the screw-drivers and has to demonstrate

[1] www.RobotCub.org



(a) (b)

Fig. 3: **The hardware used in the experiment to capture motion data.** (a) 2 wall-mounted Naturalpoint OptiTrack cameras. (b) The iCub Humanoid robot with PointGrey Dragonfly cameras.



(a) (b) (c)

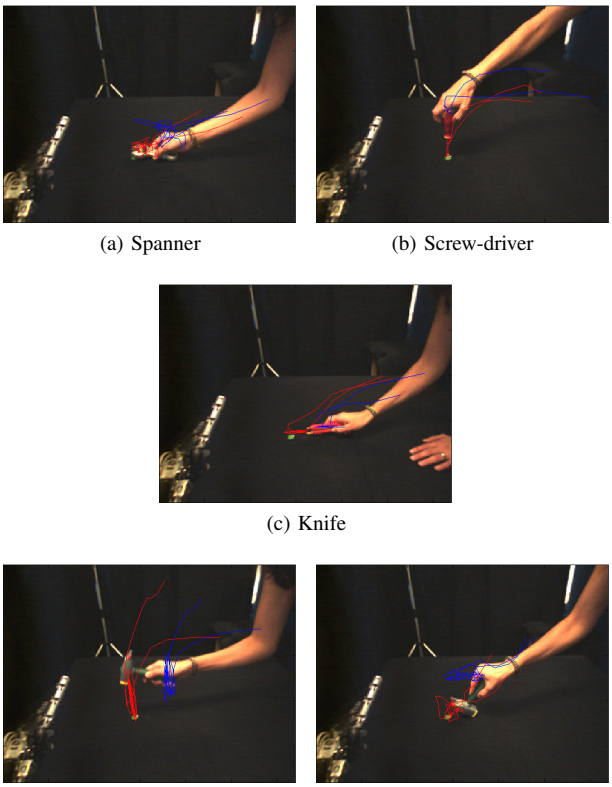
Fig. 4: **The experimental setup.** (a) the tools used in the experiment, from top to bottom screw-driver(big), hammer with claw, knife, screw-driver(small) and spanner. (b) example of the experiment with motion capture system. (c) example of the experiment with iCub in its perspective.

the use of both the hammer and the claw. The two screw-drivers with different shapes and colours and a dual-function claw-back hammer in-use are deliberately introduced to test the validity of our hypothesis that dynamical property of tools is independent of shapes and colours and to determine whether it could be applied to solve the two challenges in tool representation discussed earlier on. While each tool has an exact target to operate on in Experiment A (4, e.g. a nail for hammer and claw), Experiment B has only a pre-set target that the actions have to be effected at, which is denoted as a green patch in Fig. 4c. This is to test the performance of the model when the actions are not very precise. We then collect the samples to train our model before testing its ability to recognise an unseen tool affordance. Examples of the demonstrations for all tools are shown in Fig. 5.

B. Tracking issues

In experiment A, the average distance of all the demonstrations moved across a frame is 0.12mm. This is much smaller than the 1mm error margin of the motion capture system. Thus, we down-sampled the captured kinematics data to 10Hz before processing the data. Because of the 1mm error margin, any movement that is under 1mm across the original 10 frames is deemed stationary.

In Experiment B, We pre-mark the areas of interest on the tool template but without identifying their correspondence to the effector or handle part. The notion of these parts is identified once either of the parts gets into the target region. In this experiment, we use optical-flow technique [23] for tracking. As the features for optical-flow tracking across different instances of a demonstration vary significantly,



(a) Spanner (b) Screw-driver

(c) Knife

(d) Hammer (e) Claw

Fig. 5: **Example demonstrations of using the tools.** The red trace denotes the path of the effector while the blue one denotes that of the handle.

the centres of the areas of interest on the tools might not be a good feature to track and thus its location becomes probabilistic. To reduce the variance of the locations during demonstration, we introduced the following two measures:

1) Colour patches were introduced around areas of interest to increase local contrast and hence probability of better local tracking.
2) Gaussian smoothing technique [24] centred at the locations of interest was applied to obtain the probabilistic velocity vectors. We discretise the smoothing weights w into bins with each bin b representing 5 pixels from the centre according to (3). The number of bins is thresholded with a variance parameter σ^2 for tuning, which yields $\sigma = \sqrt{3}$ bins and $b = 4$. The weighted speed vectors are then sum and normalised against the number of tracked features in each bin and the weights to estimate the speed vector at the centre.

$$w(b) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{b^2}{2\sigma^2}} \tag{3}$$

C. Training and Evaluation

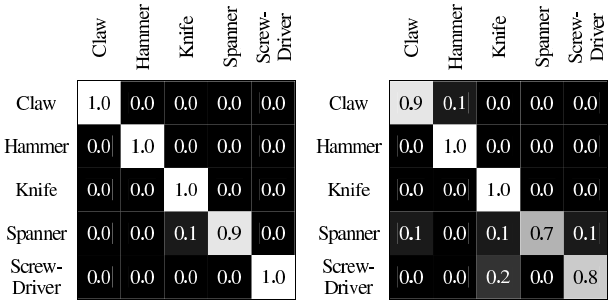
The transition probabilities and the emission distribution are estimated by iterative method to maximise expectation as discussed in [25]. We used the forward-algorithm to evaluate the log-likelihoods of a particular demonstration against the model and also the anti-models. Log-likelihoods are then

summed for all *superordinate chunks* to get the overall probability of matching. We perform "leave-one-out" cross-validation for all trials for evaluation. If the model that a given sample belongs to gives the highest log-likelihood, the sample is said to be correctly categorised. We also evaluate the performance of our strategy for automatically choosing the optimal number of states in a give HMM. The optimal number of states of a given model is compared with the number of states calculated from the automatic strategy.

V. RESULTS AND DISCUSSION

We conducted a total of 100 experimental trials, i.e. 10 sets for each experiment. This has produced 2 50×50 matrices of results for cross-validation.

Fig. 6 shows the recognition results grouped according the tools categories in the two experiments. The overall recognition rate of the model is 98% in Experiment A with accurate kinematics information, and 88% in Experiment B with imprecise target position and noise in kinematics estimation. The successful results from Experiment A confirms that our hypothesis of representing tools by their dynamical properties is valid and identification of new tool can be achieved by simply imposing a log-likelihood threshold.



(a) Experiment A (b) Experiment B

Fig. 6: **Confusion Matrix of Tool Recognition:** Columns denote models and rows denote the test demonstrations.

The only case that failed in Experiment A was a spanner being recognised as a knife. In this trial, the demonstrator accidentally rotated the spanner anticlockwise, causing the nut to be unscrewed. As the hole is significantly bigger than the bolt, from then on the entire action has shuttled about the target normal axis making it looked more like a knife cutting from many directions.

One direct result from Fig. 6 is that in both experiments, knife and hammer have been corrected identified. This can be confirmed from TABLE I which tabulates the percentage symbols present for the effector part of all tools in both experiments. We can see that the models for knife and hammers in both cases agree with each other. The one variation is that the knife in Experiment A has some contribution from rotations. This is most likely due to the demonstrators moved away either from the left or the right from the cutting slot and back to the starting point and perform the cutting demonstration again instead of simply sliding the knife up and down the slot.

The 2 mis-classification cases for screw-drivers in Experiment B happened due to the fact that some demonstrators could not keep the effector part of the screw-driver at a fixed location while rotating it because the target was just a marked point on the table. Also, without a fixed hinge for the spanner to rotate about, we observed that many demonstrators were trembling and unable to perform circular motions about the target axis. This changed the dynamic profiles that described the tool significantly which can be seen in the spanner rows in TABLE I. Although the model still classifies most trials correctly in Experiment B, we can see that this no longer captures the essence of the dynamic property of the spanner. As for the error case for claw in Experiment B, the demonstration was performed differently from the rest. Because of the missing nail for the claw to operate on, the demonstrator decided to perform significantly more nail extraction cycles repeatedly without retracting the hammer and approaching the target again.

Nonetheless, the results for Experiment B suggest that even in an unstructured environment without proper definition of a target, the dynamical properties of tools remain as a strong representation of the tools regardless of the operator.

TABLE I: The percentage breakdown of the detected velocity symbols grouped in each tool category at the effector part. The symbols are D - Departure, A - Approach, R - Rotation, S - Stationary, I - Radial, N - Normally, C - Clockwise and A - Anticlockwise.

Experiment A

	DI	AI	DN	AN	RC	RA	S
Claw	30	40	0	0	18	12	0
Hammer	2	1	47	50	0	0	0
Knife	36	33	2	1	13	15	0
Spanner	1	0	3	3	50	43	0
Screw-driver	2	1	3	4	0	0	90

Experiment B

	DI	AI	DN	AN	RC	RA	S
Claw	15	32	14	22	0	0	17
Hammer	0	0	53	44	0	0	3
Knife	51	37	4	2	0	0	6
Spanner	11	14	25	17	5	5	23
Screw-driver	9	12	24	6	0	0	49

TABLE II shows the performance of the automatic selection strategy for number of states used in our model. We can see that the number of states determined by both methods are very similar apart from the "Spanner" case in Experiment B. We can see that the λ used in the two experiments are very different, which is sensible. In Experiment A, the kinematics data is precise and accurate with target properly defined with little noise. Thus, nearly all the observations are important in describing the model. Since it is the opposite in Experiment B, we expect a smaller fraction of the data is useful.

Comparing TABLES I and II, it is interesting to observe that the optimal number of states corresponds to the number of primary abstract action symbols of the tools. This suggests that we can derive the abstract semantic description of the dynamical representation for a given tool. This piece of information, as shown in TABLE III, can be extracted from

TABLE II: The performance of the automatic strategy for selecting the number of states benchmark against the exhaustive search method.

Experiment A, $\lambda = 0.9$

Model	Claw	Hammer	Knife	Spanner	Screw-driver
Optimal	4	2	4	2	1
Automatic	4	2	4	2	1

Experiment B, $\lambda = 0.6$

Model	Claw	Hammer	Knife	Spanner	Screw-driver
Optimal	3	2	2	4	2
Automatic	3	2	2	3	2

the emissions matrices of all the tools by retrieving at every state, the dominating symbol.

TABLE III: Extracted symbolic description for the effector part of all the tools.

Model	Symbolic Description
Claw	Approach radially-Rotate anticlockwise-Rotate clockwise-Depart radially
Hammer	Approach normally-Depart normally
Knife	Approach radially-Rotate clockwise-Depart radially-Rotate anticlockwise
Spanner	Rotate-clockwise-Rotate anticlockwise
Screw-driver	Stationary

VI. CONCLUSIONS

In this paper, we argued for the need of a dynamical representation for tools. Our proposed framework, DTR is inspired by the biological evidence of the human cortex for hierarchical representations. The showcased model derived from this framework has been implemented with two plausible models in PbD, the HAMMER architecture and HMMs. Two sets of experiments have been conducted to test and verify the model for statistical significance and recognition of tools in an unstructured environment. The experimental results suggest that the model is very robust and all errors in the unstructured scenario can be accounted for systematically.

We plan to extend the work further to include dynamical description of target and also to test it with a larger set of tools. To overcome the shortfall of different tools with similar dynamic description such as a hammer and a drumstick, we will investigate on the implementation of this model together with traditional object recognition methodologies to create a hybrid framework to make the system much more robust.

VII. ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Union Seventh Framework Programme FP7/2007-2013 Challenge 2 Cognitive Systems, Interaction, Robotics under grant agreement No [270490]- [EFAA].

REFERENCES

[1] H. Murase and S. Nayar, "Visual learning and recognition of 3-D objects from appearance," *International Journal of Computer Vision*, vol. 14, no. 1, pp. 5–24, 1995.

[2] A. Berg, T. Berg, and J. Malik, "Shape matching and object recognition using low distortion correspondences," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005*, pp. 26–33, 2005.

[3] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2, pp. 2161–2168, 2006.

[4] S. Grafton *et al.*, "Evidence for a distributed hierarchy of action representation in the brain," *Human movement science*, vol. 26, no. 4, pp. 590–616, 2007.

[5] A. Martin, J. Haxby, F. Lalonde, C. Wiggs, and L. Ungerleider, "Discrete cortical regions associated with knowledge of color and knowledge of action," *Science*, vol. 270, no. 5233, p. 102, 1995.

[6] A. Martin, C. Wiggs, L. Ungerleider, and J. Haxby, "Neural correlates of category-specific knowledge," *Nature*, vol. 379, pp. 649–652, 1996.

[7] S. Nishide, T. Nakagawa, T. Ogata, J. Tani, T. Takahashi, and H. Okuno, "Modeling tool-body assimilation using second-order recurrent neural network," in *2009 IEEE Int'l Conference on Intelligent Robots and Systems. IROS 2009.*, pp. 5376 –5381, oct. 2009.

[8] C. Chen and A. Kak, "A robot vision system for recognizing 3-D objects in low-order polynomial time," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 6, pp. 1535–1563, 1989.

[9] R. Prokop and A. Reeves, "A survey of moment-based techniques for unoccluded object representation and recognition," *CVGIP: Graphical Models and Image Processing*, vol. 54, no. 5, pp. 438–460, 1992.

[10] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, pp. 511–518, 2001.

[11] P. Peursum, G. West, and S. Venkatesh, "Combining image regions and human activity for indirect object recognition in indoor wide-angle views," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 1, pp. 82–89, IEEE, 2005.

[12] E. Koechlin and T. Jubault, "Broca's area and the hierarchical organization of human behavior," *Neuron*, vol. 50, pp. 963–974, 2006.

[13] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, "Learning object affordances: From sensory–motor coordination to imitation," *Robotics, IEEE Transactions on*, vol. 24, pp. 15–26, feb 2008.

[14] E. Şahin, M. Çakmak, M. Doğar, E. Uğur, and G. Üçoluk, "To afford or not to afford: A new formalization of affordances toward affordance-based robot control," *Adaptive Behavior*, vol. 15, no. 4, p. 447, 2007.

[15] A. Stoytchev, *Learning the Affordances of Tools Using a Behavior-Grounded Approach*, vol. 4760 of *Lecture Notes in Computer Science*, ch. 9, pp. 140–158. Springer Berlin, Feb 2008.

[16] M. Kawato, Y. Uno, M. Isobe, and R. Suzuki, "Hierarchical neural network model for voluntary movement withapplication to robotics," *IEEE Control Systems Magazine*, vol. 8, no. 2, pp. 8–15, 1988.

[17] M. Nicolescu and M. Matarić, "A hierarchical architecture for behavior-based robots," in *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, vol. 1, pp. 227–233, ACM, 2002.

[18] Y. Demiris and B. Khadhouri, "Hierarchical attentive multiple models for execution and recognition of actions," *Robotics and Autonomous Systems*, vol. 54, pp. 361–369, May 2006.

[19] Y. Wu and Y. Demiris, "Hierarchical Learning Approach for One-shot Action Imitation in Humanoid Robots," in *Proceedings of the 11th International Conference on Control, Automation, Robotics and Vision. ICARCV 2010*, Dec 2010.

[20] J. R. Flanagan and R. S. Johansson, "Action plans used in action observation," *Nature*, vol. 424, pp. 769–771, August 2003.

[21] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," *Handbook of robotics*, 2007.

[22] A. Gupta and L. Davis, "Objects in action: An approach for combining action understanding and object perception," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, 2007.

[23] J. Barron, D. Fleet, and S. Beauchemin, "Performance of optical flow techniques," *International journal of computer vision*, vol. 12, no. 1, pp. 43–77, 1994.

[24] A. Wink and J. Roerdink, "Denoising functional MR images: a comparison of wavelet denoising and Gaussian smoothing," *IEEE transactions on medical imaging*, vol. 23, no. 3, pp. 374–387, 2004.

[25] S. Chatzis, D. Kosmopoulos, and T. Varvarigou, "A robust approach towards sequential data modeling and its application in automatic gesture recognition," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP 2008.*, pp. 1937–1940, 2008.