International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:3, No:10, 2009

Parametric Primitives for Hand Gesture Recognition

Sanmohan and Volker Krüger

Abstract—Imitation learning is considered to be an effective way of teaching humanoid robots and action recognition is the key step to imitation learning. In this paper an online algorithm to recognize parametric actions with object context is presented. Objects are key instruments in understanding an action when there is uncertainty. Ambiguities arising in similar actions can be resolved with object context. We classify actions according to the changes they make to the object space. Actions that produce the same state change in the object movement space are classified to belong to the same class. This allow us to define several classes of actions where members of each class are connected with a semantic interpretation.

Keywords—Parametric actions, Action primitives, Hand gesture recognition, Imitation learning.

I. Introduction

Developing human like robots is an exciting area due to its potential application in areas such as medicine [1], [2], industry [3], [4] and rehabilitation [5], [6]. One of the challenging task in building humanoid robots is to endow it with human-like capabilities. Since it is not feasible to program robots with all desired capabilities, research is being focused on building robots with learning capabilities. One approach is to teach robots new skills through demonstration where a teacher demonstrates a task and robot tries to imitate the task. Action recognition can be seen as the first step in robot task learning through imitation. With the finding of mirror neurons in brain [7], [8] , there has been a lot of attempts to enhance robot learning techniques by modeling activities with primitives [9], [10], [11]. Treating primitives as an alphabet for actions, one can think of finding a grammar for human actions [12][13][11]. This will give a unifying framework for the sensory, motor and natural language descriptions of actions. The symbolic description to actions using primitives could be thought of an intermediate level where all three descriptions meet. It is common to define the primitives manually since there is no generally accepted definition for what a primitive should be [14], [9], [15]. There has also been some attempts to find the primitives automatically so that the task would be simple [10], [16] .

Works on imitation learning fall into two groups: the first group aims at producing an exact reproduction of trajectories and the latter aims at reproducing only a subset of the predefined goals. Exact reproduction of actions are less favorable if we want the robots to have the ability to adapt to varying situations. It will not be possible to pre-program robots with all possible scenarios it might encounter. In those situations goal level imitation would be favorable which focuses on the effects of the performed action and try to reproduce those effects. In robot task learning scenarios, one usually needs to teach robots about how to handle objects. This work is considering such a scenario where a robot needs to be taught to recognize different actions that can be performed on objects in a simple table top scenario. We consider effects of manipulative actions on objects to classify actions.

Our aim is to model activities as a combination of primitive actions. In [16] a sequential learning algorithm was proposed to find primitives automatically. This approach has a limitation that the primitives are location dependent. Therefore, if the performer were to stand in a different position for the same action, a different set of primitives were found. Since the actions can take place anywhere in a scene, we need a method that will take care of the location of action/object. To achieve this, we use the concept of parametric primitives [17] to group actions with same semantic interpretation into one single group and use only one primitive to represent one group. Parametrized actions are actions that are similar but vary by some parameter θ. Parametrization allow us to specify a class of actions by some parameter θ. For example, consider pushing an object forward on a table. One can push the object at different lengths and thus we get parametrized push forward object action with parameter θ where θ represents the distance moved by the object. A modified version of the primitive extraction method in [16] is used to recognize parametrized actions.

A. Related work

Several authors have represented actions in a hierarchical manner. In [18] hierarchy of different action complexities such as *movements*, *activities* and *actions* is defined. Staffer and Grimson [19] computed a set of action primitives based on co-occurrences of observations for a surveillance scenario. In [20] Robertson and Reid present a full surveillance system that allows high-level behavior recognition based on simple actions. Their system seems to require human interaction in the definition of the primitive actions such as *walking, running, standing, dithering* and the qualitative positions (*nearside-pavement, road, driveway, etc*). These works require the manual modeling of atomic movements/primitives. We extract this kind of segmentation automatically.

The experimental results from [7] suggests that action perception and execution of motor primitives are connected through objects. There are also further studies from experimental psychology which confirms the role of objects in action understanding [21], [22]. In this paper we exploit object context to parametrize actions.

Even though object detection and classification literature is quite large (for overview see [23]), there are not many attempts to combine it with action recognition [24], [25]. In [24] Hidden Markov models are combined with object context to classify hand actions. Image, object and action-based evidence was used to label and summarize activity and also to identify objects. They define a generalized class model

to describe objects. Actions associated with each class were represented using trained HMMs. The states of such HMMs were connected to the regions through which the object moved for the particular action. A graphical Bayesian model was used in [25] for modeling human-object interactions. Some of the conditional probabilities of this model was calculated using trained HMMs. Starting and end time of action was an important feature in their model. The parametric HMM introduced in [17] indirectly model the effect of object properties with human actions. These approaches require a good initial training of action models for later recognition even though a known structure is assumed. In this work we eliminate the training task but still obtain a good model to represent and classify actions.

II. Recognition system

A. Modeling object action interactions

We consider a scenario where a robot need to be taught to recognize different actions that can be performed on a simple table top scenario. Objects are moved around the table in different directions. In this set up , we assume that at any point of time, only one object will be manipulated. To make the problem scenario more clear we give a quick summary of some of the relevant works. In [26] manipulation actions performed on a table top scenario was analyzed using hand generated primitives. They considered five actions : a) pick up an object from a table, b) rotate an object on a table, c) push an object forward, d) push an object to the side, and e) move an object to the side by picking it up. The experiments were performed by recording 3-D data using sensors attached to the body of the performer. Each action was performed in 12 different conditions: Objects placed on two different heights and two different locations on the table, and the demonstrator stand in three different locations (0, 30, 60 degrees). All the actions are demonstrated by 10 different people. Support vector machines was used to recognize manually segmented primitives and the outcome of the SVMs were fed to HMMs to model actions. The most important findings of their experiments could be stated as:a) sequences of simple semantic primitives can be used in describing actions, and b) actions learned as sequences of primitives from other demonstrators can be combined with knowledge of personal primitives to recognize new actions.

Following the above work, a method to find the primitives automatically is presented in [16]. In this work, the same data set described above was used to segment the primitives automatically in a sequential manner. Though the primitives were found automatically, the number of primitives needed to model the whole data was high since the primitives were location dependent. For each repetition, where the object or the performer was in a different position, the set of primitives were different. Even though object was present in these actions, their context was not included in the analysis. By including object context, it is possible to attach semantic interpretation to the observed primitives. For example consider the reaching of object primitive. If we are considering only the physical movement space, each time the object is moved to a different location, the reaching motion will produce different trajectory. As a result, in [16] different primitives were required to model each of these motions. But if we consider what is happening in the object space, we can summarize the whole set of reaching motion as *reaching the object*. In this sense we can group all reaching motions into one single group and then look for a representation for this group. Our argument is that actions should be grouped according to the final state of the object when the action is completed. Thus all actions that makes the same effect on the object will belong to the same action class.

When some manipulation action is performed, there is a clear segmentation of the action based on object movement:

1) Then hand is moved and reaches the object.
2) The object is moved/manipulated.
3) The hand is removed.

It is in fact what is happening in the middle that distinguishes between different actions. The classes of actions that are under consideration are such that how the action is being done is not important but what is being done is the interesting part. Hence when we want to perform classification we segment the parts where object is being moved and note its initial state and final state. When the object has stopped moving we assume that the manipulation part is completed. Depending on the final state of the object, actions will be assigned to different classes.

Our aim is to design an online algorithm where we do not have information about the number of action classes. The number different action classes will be inferred at the end.

Let $[H_t^i \ O_t^i]$ represent the feature vector for i th action sequence we are analyzing where H and O represent features for the dominating hand and object respectively. The subscript t is used for indexing time. Each of H_t and O_t are of the form $[P_t \ V_t]$ where P_t and V_t represent the position vector and velocity vector respectively. We choose t_1 and t_2 such that $|V_{t1}^O| > thresh$ and $|V_{t2}^O| < thresh$. The value of $thresh$ is chosen such that spurious movements due to measurement errors will be discarded. We can segment the sequence of observations as shown below:

$$\underbrace{H_1, H_2, \cdots}_{A}, \underbrace{H_{t_1}, \cdots H_{t_2}}_{B}, \underbrace{H_{t_2+1} \cdots H_T}_{C}$$

$$\underbrace{O_1, O_2, \cdots}_{D}, \underbrace{O_{t_1}, \cdots O_{t_2}}_{E}, \underbrace{O_{t_2+1} \cdots O_T}_{F}$$

The observations in segment E denote the part where the object is moving. Now by the transformation $O_t - O_{t_1}$ for $t_2 < t < t_1$ we can imagine that the object starts to move from the origin in each of the sequences. The transformed trajectory is then subjected to a modified version of the initial state building approach in [16] to cover the trajectory traced by the object by a sequence of Gaussians. To cover the trajectory with a number of Gaussians, we place the Gaussians in such a way that each of them cover a certain part of the trajectory. The size of the Gaussians are adjusted such that in any repetition of the same motion, each motion will have some points from each of these Gaussians. If we form a Gaussian by accumulating points over time, we might get very small Gaussian where the spacial variation is small.

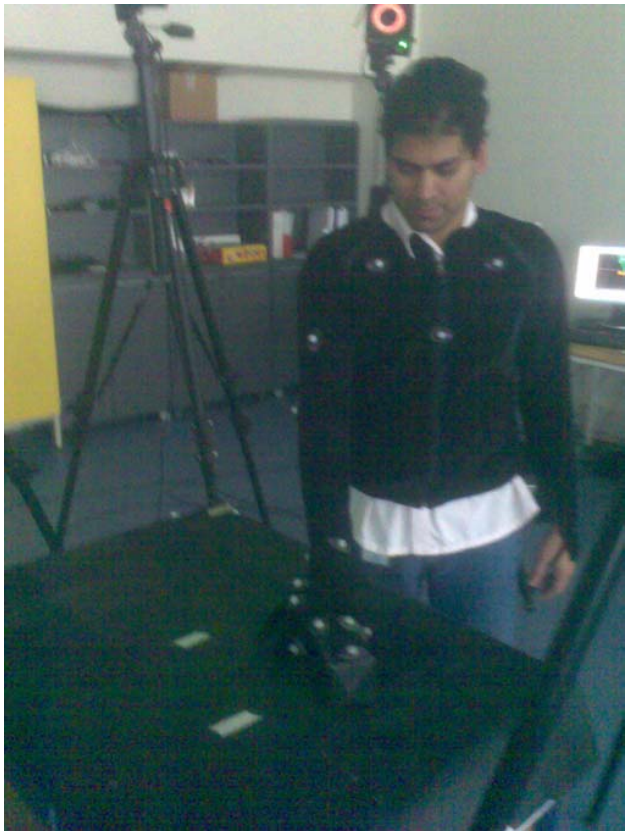

Fig. 1: The experimental setup showing the marker positions.

Therefore we look at the spatial variation to determine the placement of the Gaussians. We look at the arc length of the trajectory to determine if we have enough spacial variation to form a Gaussian. This way we avoid building a Gaussian that is too small. Our approach is outlined in Algorithm 1.

Algorithm 1: Algorithm for covering the data with Gaussians

Input: Observation sequence P=$\boldsymbol{x_1, \cdots x_t}$
Output: Means μ and Σ for a set of Gaussians covering the the trajectory
Compute $arcLen$=arc length of the trajectory
Set threshold=θ
Divide P into segments such that each part has an arc length $\lfloor arcLen/\theta \rfloor$
foreach *segment* $i =$ **do**
  Calculate $\mu(i)$=mean(segment i)
  Calculate $\Sigma(i)$=covariance(segment i)
end

Algorithm 1 is applied to each observation sequences in a sequential manner. The output of Algorithm 1 will be used to form an HMM as in [16]. A brief description of the procedure is outlined below.

Each of the Gaussians that are used to cover a sequence are then used to form a left-right HMM. We follow the notation in [27] to describe an HMM. Let $\lambda_1 = (A^1, B^1, \pi^1)$ be the initial left-right HMM for the first sequence. Set $\lambda_M = \lambda_1$. λ_M will be updated sequentially by adding more states into it or by modifying existing states. For each of the subsequent λ_k, $k > 1$, the states of λ_k and λ_M are compared. If any two states are very close, they are merged together. States that are different from those in λ_M will be added to λ_M. We use the Kullback-Leibler divergence to measure the distance between the states. When we compare two Gaussians, the Kullback-Leibler divergence has a closed form solution [28] as shown in Eqn.1.

$$D_{KL}(Q \parallel P) = \frac{1}{2}\left(\log\frac{|\Sigma_1|}{|\Sigma_0|} + \text{tr}(\Sigma_1^{-1}\Sigma_0)\right) + \frac{1}{2}\left((\mu_1-\mu_0)^T\Sigma_1^{-1}(\mu_1-\mu_0) - n\right) \quad (1)$$

Here n is the dimension of the space spanned by the random variable x, and Σ_i, μ_i represent the mean and covariance of states i=0,1.

When two states s_k, s_l are merged, all transitions to s_l are re-directed to s_k. All transitions from s_l will be adjusted to be from s_l. The output probability of the combined state will become a mixture of the output probabilities of the individual states.

When all the observation sequences have been processed, we end up with a single HMM λ_M. Sequences that belong to different class of actions will go through different sequence of states on this HMM. To find the class of the observation sequences we do the following: Each sequence is used as an observation from our HMM and the hidden states it passes through is found using the Viterbi algorithm [27]. These state sequences are then expressed as state changes by removing multiple occurrences as shown below.

$$\underbrace{s_1, s_1, \cdots s_1}, \underbrace{s_2, s_2, \cdots s_2}, \underbrace{\cdots\cdots}, \underbrace{s_k, , s_k, \cdots s_k}$$
$$\Downarrow$$
$$s_1, s_2, \cdots s_k$$

From now on by state sequences we mean the processed state sequences after removing multiple occurrences. These state-change sequences are such that sequences that with common states belong to the same class. Treating the state sequences as strings, we can pose the problem of finding the common state subsequences across various sequences as the Longest Common Substring(LCS) problem [16]. This can be solved using a dynamic program [29]. The LCS problem can be solved in O(mn) time where m, n are the lengths of strings A and B.

Once the primitives are found, we can form a primitive graph with primitives as nodes. Two nodes are connected if they appear together in some sequence. Each sequence will be a path in this tree and any two paths sharing primitives will belong to the same class. For each sequence, the parameter for the sequence will be the arc length associated with the final state.

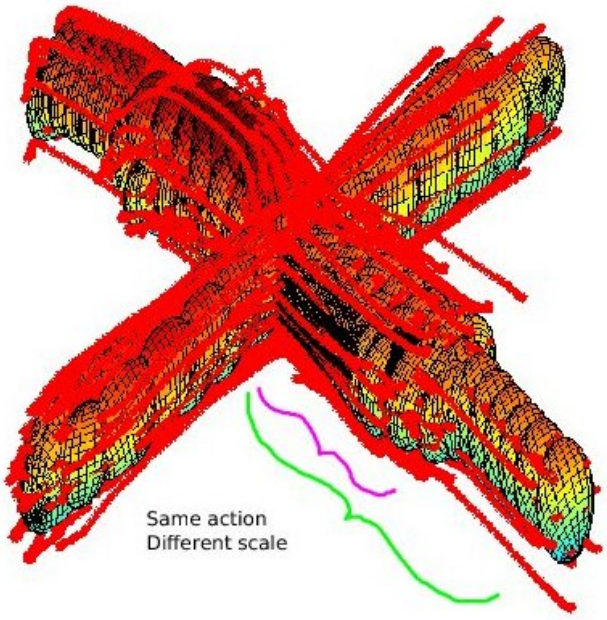

Fig. 2: Data covered with Gaussians.

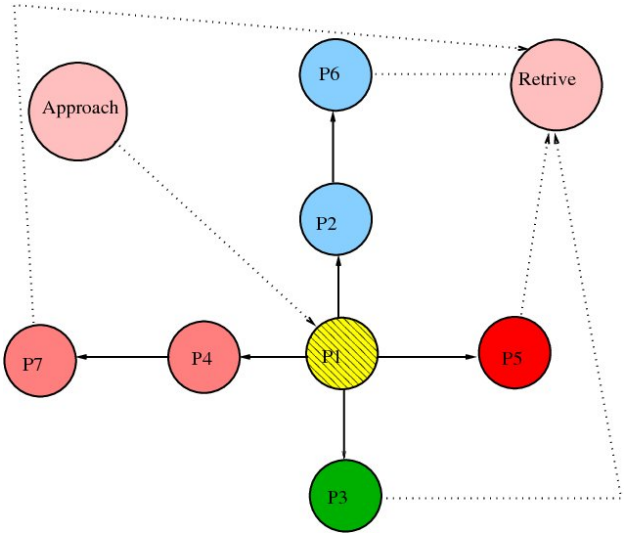

Fig. 3: Extracted primitive graph. Each path in this graph represent movements in different directions and is parametrized by the distance the object was moved.

III. Experiments

We have collected data of moving objects around on a table using a Vicon motion capture system[1]. A small box type object with 3 markers on it was moved around on a table with one hand. The hand movements were recorded using markers placed on the hand. The experimental setup with marker positions is shown in Fig. II-A. Among the 14 markers placed on the person, only the 3 markers placed on the hand was important in analyzing the actions under consideration.The recorded data contain the 3-d trajectories of hand and the object. The center of mass of the 3 markers on the hand was used as the position vector for the hand. The object was placed at different locations on the table. The object was subjected to 4 movements: push forward, push downward, push right and push left. The distance moved varied each time of the repetition. Our aim is to classify all movements in a particular direction into one class regardless of the distance the object was moved.

The recorded data was then segmented as described in Sec.II-A to find the part where the object was moved. The result of applying Algorithm.1 is shown in Fig. III. To cope with noise and variation in different sequences of same type, a few more sequences were generated by adding certain amount of noise to the input sequence before the algorithm was applied. The final grammar-like structure of the primitives found are plotted in Fig.III. In this graph, all actions are shown to start with primitive 1. But the number of observations that belong to this state are in fact very few. Few initial points are assigned to this state. Once we have enough movement to indicate the general direction of motion, the rest of the observations are assigned to appropriate states. Using the constructed model, all the actions were classified to belong to 4 classes. Each class denote movement in a This way the object movements were classified regardless of the distance it was moved.

In [16], two types of primitives were defined: those that are unique to certain actions and those that are common across different actions. Using the same primitive extraction method, we are able to represent actions of same class as a combination of some primitives. The common primitive will represent the action in the smallest scale.

IV. Conclusion

In this paper we have presented a sequential learning method for modeling and recognizing parametrized actions. Parametrized actions define a class of actions that are connected semantically through a parameter. Another advantage with our method is that all calculations are performed in the input space itself. Object context help us to group actions with a semantic interpretation. Though these initial results are not impressive they are suggestive. We plan to recognize more complex actions in our future works.

Acknowledgment

Authors would like to thank Daniel Grest for his help with recording data using Vicon system.

This work was partially supported by EU through grant PACO-PLUS, FP6-2004-IST-4-27657.

References

[1] M. A. Lewis and G. Bekey, *Automation and robotics in neurosurgery: Prospects and problems*, M. L. Apuzzo, Ed. AANS Publications, 1992.
[2] G. Ballantyne, "Robotic surgery, telerobotic surgery, telepresence, and telementoring," *Surgical Endoscopy*, vol. 16, pp. 1389–1402, 2002.

[1]http://www.vicon.com/

[3] B. Jiang, A. Sample, R. Wistort, and A. Mamishev, "Autonomous robotic monitoring of underground cable systems," in *Advanced Robotics, 2005. ICAR '05. Proceedings., 12th International Conference on*, July 2005, pp. 673–679.

[4] N. Ishikawa and K. Suzuki, "Development of a human and robot collaborative system for inspecting patrol of nuclear power plants," in *Robot and Human Communication, 1997. RO-MAN '97. Proceedings., 6th IEEE International Workshop on*, Sep-1 Oct 1997, pp. 118–123.

[5] W. Song, H. Lee, J. Kim, Y. S. Yoon, and K. Bien, "Intelligent rehabilitation robotics system for the disabled and the elderly," in *Proc. of IEEE 20th Annual International Conference on Engineering in Medicine and Biology Society*, vol. 5, 1998, pp. 2682–2685.

[6] K. Kawamura, S. Bagchi, M. Iskarous, and M. Bishay, "Intelligent robotic systems in service of the disabled," *Rehabilitation Engineering, IEEE Transactions on*, vol. 3, no. 1, pp. 14–21, Mar 1995.

[7] V. Gallese, L. Fadiga, L. Fogassi, and G. Rizzolatti, "Action recognition in the premotor cortex," *Brain*, vol. 119, no. 2, pp. 593–609, 1996.

[8] G. Rizzolatti, L. Fadiga, V. Gallese, and L. Fogassi, "Premotor cortex and the recognition of motor actions," *Cognitive Brain Research*, vol. 3, no. 2, pp. 131–141, March 1996. [Online]. Available: http://dx.doi.org/10.1016/0926-6410(95)00038-0

[9] I. S. Vicente, V. Kyrki, D. Kragic, and M. Larsson, "Action recognition and understanding through motor primitives," *Advanced Robotics*, vol. 21, no. 15, pp. 1687–1707, 2007.

[10] A. Fod, M. J. Matarić, and O. C. Jenkins, "Automated derivation of primitives for movement classification," *Autonomous Robots*, vol. 12, no. 1, pp. 39–54, 2002. [Online]. Available: http://www.springerlink.com/content/n14t2272246jj54p

[11] C. F. Gutemberg Guerra-Filho and Y. Aloimonos, "Discovering a language for human activity," in *AAAI 2005 Fall Symposium on Anticipatory Cognitive Embodied Systems, Washington, D.C*, 2005, pp. 70–77.

[12] G. Guerra-Filho and Y. Aloimonos, "A language for human action," *Computer*, vol. 40, no. 5, pp. 42–51, 2007.

[13] A. Kojima, T. Tamura, and K. Fukunaga, "Natural language description of human activities from video images based on concept hierarchy of actions," *Int. J. Comput. Vision*, vol. 50, no. 2, pp. 171–184, 2002.

[14] O. C. Jenkins, M. J. Mataric, and S. Weber, "Primitive-based movement classification for humanoid imitation," 2000.

[15] M. J. Matarić, B. Zordan Victor, and M. M. Williamson, "Making complex articulated agents dance," *Autonomous Agents and Multi-Agent Systems*, vol. 2, no. 1, pp. 23–43, 1999.

[16] Sanmohan and V. Krueger, *Primitive Based Action Representation and Recognition*, ser. Image Analysis, R. J. A.B Salberg, J.Y. Hardeberg, Ed. Springer Berlin / Heidelberg, 2009, vol. LNCS 5575. [Online]. Available: http://www.springerlink.com/content/yv5501516404t5h8/?p=d1f1d06139ae4f05ba5d6701fd9a4e2c&pi=3

[17] A. D. Wilson and A. F. Bobick, "Parametric hidden markov models for gesture recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, pp. 884–900, 1999.

[18] A. Bobick, "Movement, Activity, and Action: The Role of Knowledge in the Perception of Motion," *Philosophical Trans. Royal Soc. London*, vol. 352, pp. 1257–1265, 1997.

[19] C. Stauffer and W. Grimson, "Learning Patterns of Activity Using Real-Time Tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 747–757, 2000.

[20] N. Robertson and I. Reid, "Behaviour Understanding in Video: A Combined Method," in *Internatinal Conference on Computer Vision*, Beijing, China, Oct 15-21, 2005.

[21] K. Nelissen, G. Luppino, W. Vanduffel, G. Rizzolatti, and G. A. Orban, "Observing Others: Multiple Action Representation in the Frontal Lobe," *Science*, vol. 310, no. 5746, pp. 332–336, 2005. [Online]. Available: http://www.sciencemag.org/cgi/content/abstract/310/5746/332

[22] D. Bub N and Michael E. J. Masson, "Gestural knowledge evoked by objects as part of conceptual representations," *Aphasiology*, vol. 20, no. 9-11, pp. 1112–1124, November 2006. [Online]. Available: http://dx.doi.org/10.1080/02687030600741667

[23] S. Ullman, "High-level vision: Object recognition and visual cognition," July 1996.

[24] D. Moore, I. Essa, and I. Hayes, M.H., "Exploiting human actions and object context for recognition tasks," vol. 1, pp. 80–86 vol.1, 1999.

[25] A. Gupta and L. Davis, "Objects in action: An approach for combining action understanding and object perception," pp. 1–8, June 2007.

[26] I. S. Vicente, V. Kyrki, and D. Kragic, "Action recognition and understanding through motor primitives," *Advanced Robotics*, vol. 21, pp. 1687–1707, 2007.

[27] L. R. Rabiner, "A tutorial on hidden markov models and selected applications inspeech recognition," *Procceeding of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[28] Vladimir Dragalin, V. Fedorov, S. Patterson, and B. Jones., "Kullback-leibler divergence for evaluating bioequivalence," *Statistics in Medicine*, vol. 22, no. 6, pp. 913–930, 2003.

[29] D. S. Hirschberg, "Algorithms for the longest common subsequence problem," *J. ACM*, vol. 24, no. 4, pp. 664–675, 1977.