Mealy Machine Implementation of a Humanoid-Oriented Movement Writing

Adrian Stoica
NASA Jet-Propulsion Laboratory
University of California Berkeley United States
adrian.stoica@jpl.nasa.gov

Kelvin M. Liu-Huang
Carnegie Mellon University
kmliu@andrew.cmu.edu

Hyung Ju Suh
California Institute of Technology
hsuh@caltech.edu

Sarah Martin
Arizona State University
sarahmartin@asu.edu

Steven M. Hewitt
University of California Berkeley
steven.hewitt.cal@gmail.com

Sarah Bechtle
California Institute of Technology
sbechtle@caltech.edu

Abstract:

The humanoid-oriented movement (HOM) writing system was recently introduced in [1] as a natural modality for encoding the movements that humans or humanoid robots perform during various work activities. Inspired by Sutton Movement Writing and Shorthand [2], HOM Writing depicts movement using body postures, which are easily visually interpreted by both humans and humanoid robots. Humanoid robots could directly map the key postures represented in the notation to their own postures, imitating the postures captured in the description, computing intermediate postures, and interpolating optimal paths to create continuous movements. In this paper, we extend HOM by representing humanoid activity as a Mealy machine [3], which lets the humanoid respond to the environment, make decisions, and perform repetitive behavior. We discuss how HOM, a compact and intuitive humanoid language, allows quick non-expert robot programming, buildup of humanoid movement knowledge, and semi-supervised learning of motion planning. We also developed a process to automatically extract *poses* (humanoid postures represented by joint angles) from movement sequences found in video recordings of humanoid activity using the pose estimation system described by Yang and Ramanan [4]; and describe a means of converting movement sequences into HOM diagrams. This paper specifies the HOM System as a formal language, and demonstrates an algorithm for translating a movement *scene*, wherein the humanoid performs repetitive movements and responds to an environmental stimulus, from video into HOM writing.

Keywords: Movement Writing, Humanoid, Mealy Machine

1st Conference on Robot Learning (CoRL 2017), Mountain View, United States.

1 Introduction

Automation has greatly increased quality of life, and has suffused all facets of human society in the past decades. The last frontier of automation is building adaptable automata through computer programming, but progress is slow due to excessive reliance on highly educated programmers. Furthermore, current software is typically platform-specific, which is problematic because platforms and language specifications have high turnover. Effective visual programming languages, which increase accessibility of programming, e.g. the Microsoft Visual Programming Language [5], are becoming increasingly available but are still insufficiently expressive and intuitive for representing humanoid movements.

Robot learning is essential for quickly, accessibly, and adaptably transferring skills to robot apprentices. While programming through numerical (sub-symbolic) representation has been predominant, using symbols for internal representation as well as human-like communication is an important alternative. Learning like humans through compact and intuitive communication would provide humanoid robots, in the same way it provides humans, a rapid way to learn humanoid behavior. However for communicating actions, natural language is too complex and imprecise, while also not the most expressive medium for describing body movements. Therefore we describe a humanoid-oriented movement (HOM) writing system to provide a compact and expressive means to communicate humanoid movement behaviors.

Besides expediting robot programming, HOM also provides a mid-end abstraction of movement in a learning infrastructure. On the front-end, a humanoid robot observing humanoid behavior, either in real-time or recorded video, can easily transcribe the behavior into HOM scripts. Then, that robot can read the script and learn to imitate the behavior through back-end learning or interpolation of motion control. These HOM scripts can even be understood and refined by any humanoid if necessary. Using this process, even non-programmers could teach movement behaviors to robots. This is particularly important out in the field, especially remote and dangerous locations such as space or battlefield, where the life of a programmer might be at risk and yet the robot needs to be flexibly programmed/instructed.

Going a step further, HOM scripts can be compactly archived and shared through a database, which would improve buildup of humanoid knowledge, eventually leading to a complete movement lexicon. Further coupling HOM scripts with speech recognition and learned motion control would help the humanoid query HOM scripts from the database through semi-supervised learning, and adapt them perform novel behaviors. Such a multimodal and distributed representation of movement could be interpreted as improved "understanding" because it increases the robot's accuracy in solving movement tasks. Some even theorize that such an embodied understanding of the world is *required* for human-like intelligence [Stoica, 1997].

2 Previous Work

The use of movement writing for robotics was proposed in the early 90s [6], [7]. Virtually all work that followed [8], [9], [10], [11] used the most popular movement writing systems in dance and choreography, the Laban notation [4], which provides a complete language for movement description.

In [8], Knight and Simmons adapt the Laban notation [12] to the movement of a 2-DOF Aldebaran Nao head and a 4 DOF Keepon. The robot movement features were created manually, and no automatic movement detection was used. The study focused on determining how well users of Amazon Mechanical Turk[4] would correctly interpret the robot movement. Samadani et al. [9] describe how they adapted two existing quantification approaches of Laban components for hand and arm movements only. Six hand and arm motion paths were designed to convey six basic emotions. The hand and arm movements were Laban annotated by a certified movement analyst (CMA) to compute the statistical correlation between the CMA-annotated and the quantified Laban components. The results show that the correlation between the CMA annotated and the quantified outcome is high (˜80%). In [11] the authors present a framework for emotion recognition from video. The hand of a person was tracked, and the analysis of the tracking showed that acceleration and frequency characteristics of the hand are relevant to recognizing emotion. The authors argue that a computer encoding of the Laban movement can serve as a common language for expressing and interpreting

emotional movements between robots and humans. In [10] Hachimura et al. again compared the Laban movement notation extracted algorithmically from a motion capturing system with the results of the analysis of a specialist. They achieve partially satisfactory results from the comparison but claim that a numerical formulation of the Laban movement notation is possible. The work presented in [13] and [14] formulates solutions for retargeting human motion to humanoid motion, which is of interest when adapting any movement writing system to robotic motion. In [13] human motion was captured by a motion capture system and then converted to humanoid movement. In [14], the human motion data was obtained from a human motion database and a pose tracking system, but only the human upper body motion was retargeted to the robot.

These previous attempts to retarget human motion to humanoid motion provide evidence to the interest in a robot imitating or following human movement. Nevertheless, the work in [13] and [14] fall short of formal writing system that generalizes movement representation. The majority of the previous work focuses on the Laban notation, and rarely considers full anthropomorphic body movement. Also, the Laban system is visually less intuitive (to those unfamiliar with the notation), and cannot be directly translated into movement, compared to Sutton notation [2].

In [1] we introduced HOM Writing, a humanoid-oriented movement writing based on the Sutton notation and we argued it is a system better suited for notation of humanoids movements. In that paper we introduced a HOM editor and a demonstration of automatic posture capturing from video, wherein postures were transcribed directly from videos of human activities to HOM Writing. Whereas in the past, most approaches to transcribing movement relied heavily on motion capturing technology.

HOM v.1 [1] was limited to expressing only fixed sequences of movements/operations. While this limitation is sufficient for dance choreography, which usually seeks to express a predetermined sequence of movements for artist purposes, robotic movement is typically used for practical purposes. Therefore our writing system needs to be able to encode repetition and interaction with objects (which can be expressed in Sutton notation), and conditional behavior (i.e. regular grammar). Besides repetition and conditionality, having unlimited memory (e.g.unrestricted grammar) would also be desirable, but for reasons we will discuss later it may be impractical to include.

In this paper we propose HOM v.2, expanding the capabilities of the language to a Mealy finite-state machine (or Mealy machine). This relatively minor modification greatly increases the capability of the language, allowing it to encode repetition and conditional behavior. Furthermore, we show that this improvement requires few changes to the user interface and sacrifices little intuitiveness and usability.

3 HOM Writing System

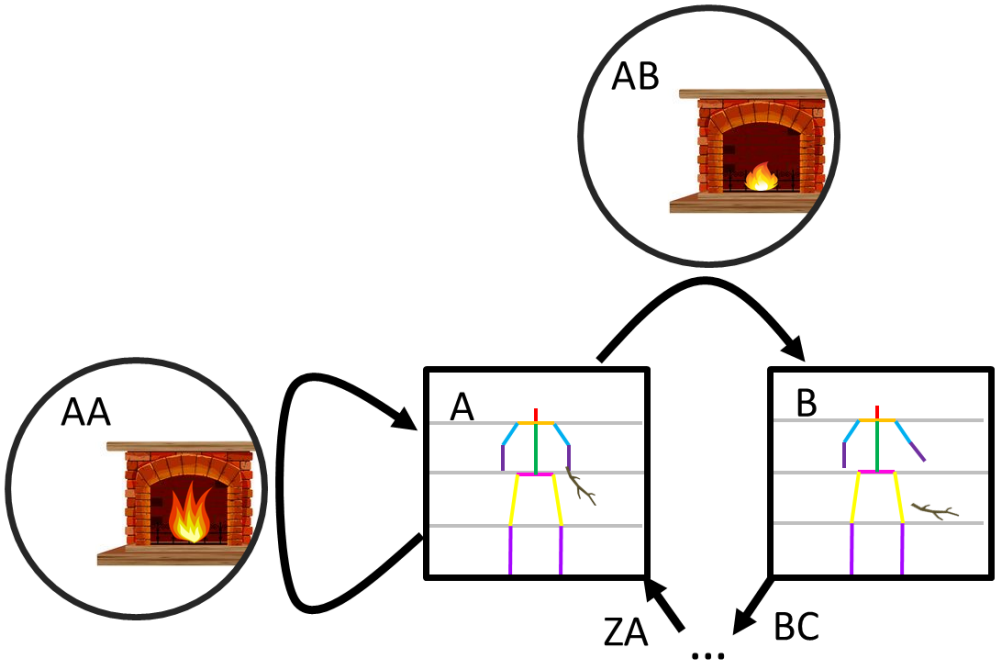

Figure 1: Example HOM diagram for a fire-stoking humanoid. Assuming the humanoid begins in pose A with a piece of wood in hand, the humanoid then checks the environment to decide how to proceed. If the fire is sufficiently large, the humanoid takes transition AA and returns to pose A. If the fire has run low, the humanoid takes transition AB, dropping the wood into the fire. From pose B, the humanoid then proceeds through an additional sequence of states (which for simplicity are not shown) to acquire more wood. For example, there might simply be a cache of wood next to the fireplace that the robot simply walks over to retrieve.

3.1 HOM v.2

In HOM v.2, we abstractly represent humanoid behavior as an embodied automaton with computational power equivalent to a *deterministic finite automaton* (DFA). As later discussed, our abstraction more specifically resembles a Mealy machine. An automaton is a system which always resides in one particular internal *state* at any given time (Fig. 1), out of finitely many states predefined by the programmer. For HOM in particular, each internal *state* corresponds to a physical posture, or *pose*. Multiple states can produce the same pose, but one particular state always produces the same pose (i.e. poses are a many-to-one function of states).

Similar to Sutton notation, humanoid posture is represented by an anthropomorphic stick figure because stick figures are easy to visualize and interpret. Unlike Sutton notation, we modified the stick figures to allow only straight segments, corresponding to robot links, as well as linearized rigid bone formations. HOM inherits all the conventions of the Sutton Writing notation, but further adaptation to suit robot bodies and robotic activities are needed. HOM v.2 includes the following elements visible in Figure 1:

1. Each pose depicts the posture of the humanoid being controlled (the actor). It does not contain any other objects or environmental features.
2. The actor is represented by eleven line segments, which indicate: The head (one line), the shoulders (one line), the spine (one line), the upper arms (two lines, one line per arm, the

forearms (two lines, one line per arm), the hips (one line), the thighs (two lines, one line per leg), the lower leg (two lines, one line per leg)

3. Like a musical staff, five horizontal lines specify the normal position for the foot line, the knee line, the hip line and the shoulder line. The most upper line specifies the height of the arm when lifted over the head.

The humanoid switches between states by taking one of the *transitions* (arrows) leading from one state to another. From the same initial state, the humanoid may transition to a different state under a different condition (Fig. 1) (i.e. given a different input stimulus). Rather than a low-level representation of input (i.e. an array of sensors or conditions), HOM uses a high-level visual representation of the environment as the input condition (Fig. 1). The transition condition can include any or all relevant environmental features or objects. Like the pose, we would use similar abstractions to model environmental objects (e.g. crates are brown cubes, the wall is a textured surface). Whether we use a feature-based, parts-based, or holistic model of the environment, the humanoid simply matches the environment to the most similar condition for transition and performs that transition.

Different arrangements of states and transitions in this flowchart diagram (called a *scene* in HOM) produce different behaviors. Also, physical movement do not necessarily occur during these transitions, because transitions can also be used to perform internal computation (like a flowchart) rather than take actions. Thus, the writer can create humanoids which respond in innumerable different ways (though not as many as a Turing machine) to the environment through careful design of the *scene*.

3.2 Abstract Machines / Formal Languages

Deterministic finite automata (DFA) are machines which can only produce "accept" or "reject" (or "yes" or "no") as output. Thus they can only answer "yes" or "no" questions, called "decision problems". We instead need to solve "function problems" because we expect humanoids to perform a variety of tasks, rather than binary tasks. A Mealy machine achieves this by producing various outputs (in our case, movements), rather than binary outputs, during each transition. Meanwhile, Moore machines, which produces output upon reaching states rather than during transitions, are safer to use because asynchronous feedback may occur during interactions between multiple Mealy machines. However, Mealy machines react faster due to the same lack of synchronicity and tend to have fewer states and transitions. Since our primary goal is currently intuitiveness, we base HOM on Mealy machines.

Due to the equivalence in computational power between a Mealy machine and a DFA, the humanoid behaviors described by HOM can be generally called finite automata. In fact, any Mealy machine can be simulated by a sufficiently large number of separate DFAs. Thus each HOM script is a regular language, and the set of all HOM scripts comprise a regular grammar.

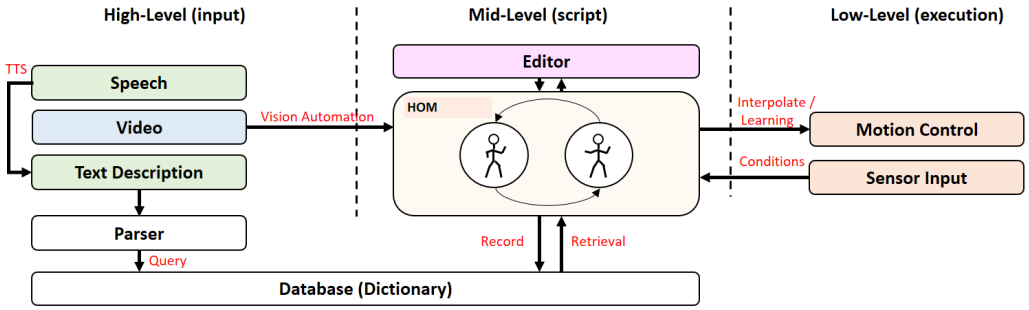

Figure 2: Schematic diagram of the abstraction layers of HOM Writing, along with existing and future tools and modalities that interact with HOM

3.3 Layers of Abstraction

We previously identified a gap between front-end communication of movement, in the form of command phrases and learning by observation, and back-end motion control, which is still typically manually programmed. As an abstract language, HOM Writing serves to bridge the the front- and back-end, mirroring low-level, back-end robotic components such as control variables.

Through pose estimation of video or motion capture followed by causal inference, a humanoid can transcribe movement behavior into HOM scripts. Additionally, associating phrases describing movement (e.g. captured through speech recognition) with the HOM scripts would let humans directly teach, label, and describe movement routines to robots in a very high-level manner (i.e. this is "walking" or "picking up"). This mirrors the way a human who watches an instructional video extracts and names salient poses and movement primitives, abstracting away the details of motion control which are too specific for different scenarios. Compared to directly learning motion control from video, learning HOM scripts grants the additional benefit of allowing writers to understand, refine, and reuse recorded scripts. Our HOM editor allows even novice programmers to manually draw or refine poses.

HOM itself is intended only as a high-level representation of movement task understanding, with the goal of being compact and intuitive for model portability and human editors. Poses are not meant to encompass the gamut of nuanced motions required, for example, to catch an animal, which would be innumerable. They are intended as a high-level outline of the key motion primitives required to perform a task. Complex motion control would be offloaded from human programmers to humanoid robot automation. Rather than manually programming low-level, back-end motion control, the humanoid learns these through reinforcement learning or simulated evolution. Since HOM scripts only describe salient poses, the humanoid must learn how to transition between them. Each transition is a separate learning problem wherein the humanoid must learn how to move from the predecessor pose to the successor pose. The humanoid stochastically explores trajectories to achieve the movement task, and successful trajectories are rewarded or recorded.

Most importantly, archiving and sharing HOM scripts through a database greatly expands the power and accessibility for humanoids to learn movement behavior. Unlike problem-specific, platform-dependent programming, archiving HOM scripts allows any humanoid to unlimitedly reuse and adapt scripts learned by a particular humanoid, building up a lexicon of movement knowledge. Additionally, archiving scripts labeled with speech (or other sensorial input) would allow other humanoids to retrieve scripts through natural language description, and even construct new scripts through semi-supervised learning. Finally, motion control would also be archived along with HOM scripts, allowing other humanoids querying from the database to use semi-supervised learning to adapt motion control for novel movement tasks. Even combining simple scripts in these diverse way would allow humanoids to perform complex tasks, or *stories*.

4 Transcribing Motion from Video to HOM : Case Analysis

4.1 Transcription Process

In this section we explore the possibility of using videos as inputs to HOM, and the process of automating video inputs to HOM notation. In this specific scenario, HOM serves as an analogy for an abstract thought process that humans would go through between watching an instructional video of motion, and executing the motion. The process may happen subconsciously, but it is imperative that some key gestures/motions are remembered from this video, which we may later interpolate to carry out the same motion that we have perceived.

From this concept we can immediately identify two tasks that are necessary in order to abstract, and reconstruct the motion from a video: the extraction of skeletons (feature matching to perceive motion information from the video), and the extraction of motion primitives (key points in time that are integral to interpolation). The approach taken is mostly similar to the work in [1], except that we expand the script notation before to a more compact and capable state diagram.

In [1], joint extraction (skeletonization) is done using software by Yang and Ramanan [4]. The work of Yang and Ramanan uses learning on pose databases, and strengthens the learning process by inferring the position and orientation of joints using their locality. The first proposed HOM [1]

attempts to use this algorithm on videos instead of images, and attempt to represent motion points by conformally normalizing (preserving angle between joints) the skeleton.

While [1] was successful in recovering angular features from videos, this approach is limited in its capability to extract key motions due to a high noise in joints where these key features likely occur. For example, while the torso joint will provide a low-noise, it is unlikely to represent key points of the motion. On the other hand hand and leg joints are much more meaningful, but are too noisy for analysis.

We use the same process to extract the skeletons and the angle data of a video, but instead compare this to the original HOM notation to illustrate the compactness and extended capability of the new HOM notation. The scenario to be tested is simple, and involves a human walking and picking up a cone, and resuming walking.

4.2 Conversion to HOM Diagram

From the video transcription, we obtain a sequence of joint angle vectors, also called actor vectors. Let actor matrix **A** be an *N* x *T* matrix where *N* is the number of joints and *T* is the number of frames in the video. We can treat these actor vectors as coordinates in n-dimensional space (in this case nine-dimensional due to having nine joints). Therefore *movement* is a trajectory through this n-dimensional joint space. During movement, the environment is changing as well. Since we assumed the environment is also described by a feature vector (see section HOM v.2), we can represent the environment **E** over time as an *M* x *T* matrix, where *M* is the size of the environment vector.

Given the movement and environment trajectories, we need to learn a HOM diagram that estimates the humanoid's behavior. This is essentially a causal inference problem because we want to predict the current actor vector from the past actor and environment vectors. To begin, we need to find the regressor $f(\boldsymbol{a}_{*1}, ..., \boldsymbol{a}_{*,t-1}, \boldsymbol{e}_{*1}, ..., \boldsymbol{e}_{*,t-1})$ which minimizes the mis-classification error $\frac{1}{T}\sum_{i=1}^{T}(f(\boldsymbol{a}_{*1}, ..., \boldsymbol{a}_{*,t-1}, \boldsymbol{e}_{*1}, ..., \boldsymbol{e}_{*,t-1}) - \boldsymbol{a}_{*t})^2$. This graphical model inference is already a computationally intensive task, but it does not directly produce the desired HOM diagram because it encodes too much noise. The raw regression would produce spurious results such as relationships between current states and states in the distant past, unlikely to be related. Additionally it infers the optimal outputs over the entire domain of the regressor, whereas our HOM diagram is only interested in high-confidence relationships. Therefore, during or after the inference, we need to perform additional dimensionality reduction or anomaly detection in order to identify sparse, high-confidence solutions/relations. Each solution/relation, represented by a sparse matrix of the form $(\boldsymbol{e}_{*1}, ..., \boldsymbol{e}_{*,t-1}, \boldsymbol{a}_{*1}, ..., \boldsymbol{a}_{*,t-1}, \boldsymbol{a}_{*,t})$, can be represented as a partial HOM diagram. Then we merge all the solutions into a complete HOM diagram by matching identical or similar pose sequences, which is relatively simple compared to the previous two tasks.

Since the aforementioned graphical modeling and dimensionality reduction tasks are still open problems, we hope to address them in future work, or using future advancements in machine learning.

5 Results

Using the process described in the Transcribing Motion from Video to HOM section, we skeletonized human figures from individual frames of videos (Figure 3) by modifying the software from Yang and Ramanan[4]

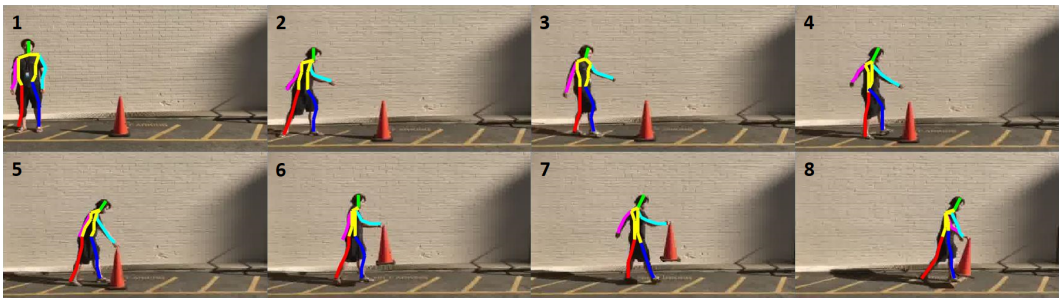

Figure 3: Extracted skeletons from scene where a human walks, picks up a cone, and resuming walking, using software from Yang and Ramanan[4]

Then, we automatically extract joint angles from the segmented sequence, and conformally normalize this skeleton by transcribing the angle data into the HOM editor:

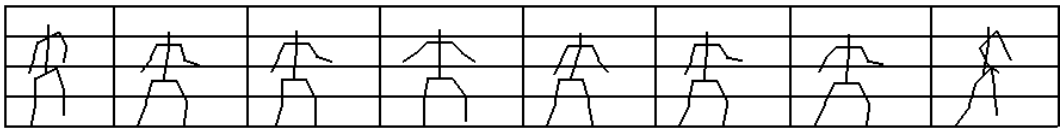

Figure 4: Conformally normalized skeleton transcribed into the HOM v.1 editor

In HOM v.1, the whole sequence can be represented with the following pseudocode:

Algorithm 1 Script Algorithm for simple scene of walking and picking up a cone

```
1:  procedure WALK_AND_PICK_UP_CONE
2:      Stand Front
3:      Step left
4:      Step right
5:      Step left
6:      Pick up cone
7:      Step left
8:      Step right
9:      Step left
10: end procedure
```

We can see that HOM v.1 is essentially a scripting language without support for if statements and for loops. This prevents the language from specifying decisions and complex motions in response to environmental conditions (from sensor input), and forces us to describe lengthy repetitive movements statement by statement.

By representing the scripting process with a state diagram, HOM v.2 allows for more compact and flexible programming, while maintaining the symbolic clarity of HOM v.1. For the previously described scene, by applying Algorithm 1 to the movement sequence, we obtain the following state diagram:

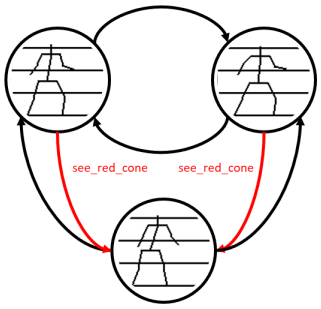

Figure 5: Repetitive and conditional behavior state-based HOM Writing System

The state diagram in Figure 4 can be described using the pseudocode in Algorithm 3.

Algorithm 2 State-based Algorithm for simple scenario of walking and picking up a cone

```
1:  procedure WALK_AND_PICK_UP_CONE
2:      Stand Front
3:      while True do
4:          Step left
5:          Step right
6:          if see red cone then
7:              go to 11
8:          else
9:              go to 4
10:         end if
11:         Pick up cone
12:     end while
13: end procedure
```

By comparing Algorithm 2 and Figure 5 against Algorithm 1 and Figure 4, we can see that the use of state diagrams makes HOM v.2 more compact and intuitive to humans. For scripts that include long repetitive actions, response to conditions, or decision/inference problems, HOM v.2 represents the behavior much more effectively and intuitively than HOM v.1.

6 Conclusion

In this paper, we have described an enhancement to HOM Writing, our previously introduced movement writing system tailored towards humanoids. We have formalized HOM v.2 by expressing humanoid behavior as a Mealy machine, which allows it to respond to stimuli, make decisions, and perform repetitive behavior. We have described an automated method for transcribing humanoid movement from a video to a motion script and the means of converting scripts to HOM state diagrams. We successfully extracted a sequence of poses from a video of human movement, and converted the sequence into a HOM diagram manually. As can be seen, the HOM diagram is simpler than the full motion sequence, and better corresponds to how humans recall movement; thus the method presented is intuitive, flexible, and compact in describing motion sequences. As currently described, the algorithm may be an intractable problem which can be addressed in future work.

The uses of HOM are not limited to the simple case analysis illustrated in this paper. Coupling movement scripts with speech recognition, sensorial input, and learned movement control would provide a multimodal representation of movement, which could be interpreted as improved "understanding" because it increases the humanoid's ability to solve novel movement tasks through semi-supervised learning. Furthermore, storing these scripts to a large database, would allow humans and humanoids to share and understand movement tasks, leading to a buildup of transferred knowledge rather than wasting time reprogramming robots with the same motion control routines.

Acknowledgments

To be included later.

References

[1] A. Stoica, H. J. Suh, S. M. Hewitt, S. Bechtle, A. Gruebler, and Y. Iwashita. Towards a humanoid-oriented movement writing. *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2017.

[2] V. Sutton. *Sutton Movement Shorthand: Book I: The Classical Ballet Key*. Movement Shorthand Society, 1973.

[3] G. H. Mealy. A method for synthesizing sequential circuits. *Bell Labs Technical Journal*, 34 (5):1045–1079, 1955.

[4] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1385–1392. IEEE, 2011.

[5] K. Johns and T. Taylor. *Professional microsoft robotics developer studio*. John Wiley & Sons, 2009.

[6] T. Nakata, T. Sato, T. Mori, and H. Mizoguchi. Expression of emotion and intention by robot body movement. In *Proceedings of the 5th international conference on autonomous systems*, 1998.

[7] J. T. Butler and A. Agah. Psychological effects of behavior patterns of a mobile personal robot. *Autonomous Robots*, 10(2):185–202, 2001.

[8] H. Knight and R. Simmons. Laban head-motions convey robot state: A call for robot body language. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 2881–2888. IEEE, 2016.

[9] A.-A. Samadani, S. Burton, R. Gorbet, and D. Kulic. Laban effort and shape analysis of affective hand and arm movements. In *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on*, pages 343–348. IEEE, 2013.

[10] K. Hachimura, K. Takashina, and M. Yoshimura. Analysis and evaluation of dancing movement based on lma. In *Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE International Workshop on*, pages 294–299. IEEE, 2005.

[11] T. Lourens, R. Van Berkel, and E. Barakova. Communicating emotions and mental states to robots in a real time parallel framework using laban movement analysis. *Robotics and Autonomous Systems*, 58(12):1256–1265, 2010.

[12] V. Laban. *Labans Practice of Dance and Movement Notation*. Princeton Book Co Pub, 1975.

[13] S. Kim, C. Kim, and B.-J. You. Whole-body motion imitation using human modeling. In *Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference on*, pages 596–601. IEEE, 2009.

[14] B. Dariush, M. Gienger, B. Jian, C. Goerick, and K. Fujimura. Whole body humanoid control from human motion descriptors. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 2677–2684. IEEE, 2008.