Moving in Virtual Space:
A Laban-inspired Framework for Procedural Animation

Swar Gujrania, Duri Long, Brian Magerko
Georgia Institute of Technology, 85 5th St. NW Atlanta, GA 30309
{swar.gujrania, duri, magerko}@gatech.edu

Abstract

Procedural content generation has been widely used for aiding game content designers in the content creation process. However, limited work has focused on autonomously generating novel physical gestures for non-player characters. This paper focuses on using creative strategies from theater and dance to enable the generation of novel movement sequences for characters in 3D game environments. We apply the Space component from Laban's Movement Analysis to procedurally improvise movements in virtual environments. Our framework outlines a pipeline for an improvisational AI that analyzes human movements in terms of the geometric properties of Laban's icosahedron and manipulates these projections to generate various alternative movements. In this paper, we mainly focus on describing the structural design and encoding scheme used in the framework.

Introduction

Games are a massive and continuously growing industry (Hendrikx et al. 2013). This growth, however, creates an increasing need for novel game content, which can be time-consuming and expensive to produce manually (Hendrikx et al. 2013). Procedural Content Generation (PCG) can provide a solution to this issue. PCG is defined as the computer-aided generation of game content that uses a random or pseudo-random process to generate different gameplay spaces (Pcg.wikidot.com, 2019). PCG reduces the burden on game content designers by aiding them in, or in some cases even taking over, the content creation process. A wide body of research has investigated the developing and applying new PCG techniques for game components (Hendrikx et al. 2013). There has also been a lot of research on creating increasingly natural, expressive and believable non-player character (NPC) animations using PCG. The NPCs are also becoming increasingly intelligent in their ability to learn from player behavior and understanding of game context (Yildirim and Stene 2008).

An important component of intelligence is autonomy in actions (Brooks 1991). While there is some research on procedurally generating prespecified gestures for humanoid characters (Johansen 2009, Albrecht et al. 2006, Horswill 2009), little work has been done on generation of novel gestures for NPCs (Togelius et al. 2013). Making NPCs more autonomous could enable characters to respond to novel situations and unexpected inputs in a more natural way, creating a more dynamic and engaging game environment. Content generators that draw on theories of creativity can be used to generate more "coherent, original, and creative" content, but this is a space which needs more research (Togelius et al. 2013).

This paper focuses on using creative strategies from theater and dance to enable the generation of novel movements for characters in 3D game environments. More specifically, we apply Laban's Movement Analysis (LMA) (Laban 1970, Laban 1966) to the development of procedural movement improvisation techniques in virtual environments. This includes applications like procedural character animation in games and creativity support tools for animators and choreographers. LMA's structured nature and flexible granularity of analysis makes it a robust framework for computational purposes. Previous works have employed LMA for movement analysis and stylization (e.g.: Dias 2007, Mentis and Johansson 2013), but we are specifically looking at movement generation. The *Space* component of LMA, in particular, lends itself well to reasoning about abstract movements in terms of geometry (Sutil 2013) and has been largely uninvestigated in the literature on computational creativity and procedural animation.

Our larger research agenda is to develop an AI system that can reason about human movements to establish a deeper

---

computational understanding of human behavior. In this paper, we take a step towards realizing that goal by presenting a theoretical framework for procedural animation using the *Space* component of LMA. We outline a pipeline for an improvisational AI that can analyze human movements in terms of the geometric properties of Laban's icosahedron and manipulates these properties to direct the generation novel alternative movements. This work lays ground for an alternative method for reasoning about human movements, which can then be applied to real-time interaction in games using full-body gestures (e.g. VR or motion-capture based games), or even expanding the gesture set of a humanoid avatar from a limited set of predefined animation clips.

Related Work

Over the years, a lot of research has gone into PCG for games (Hendrikx et al., 2013, Yildirim and Stene 2008, Togelius et al. 2013). Many research efforts have tried to simulate components of human behavior--particularly gestures--in the past (Shapiro 2011). Some systems focus on gesture generation (Johansen 2009), others focus on creating movements that are more believable, expressive and natural (Chi et al. 2000, Bleiweiss et al. 2010, Miyake 2015). A variety of research has taken inspiration from the performing arts to impart a style, mood, or personality to a movement (Neff and Fiume 2003, 2004, Maraffi and Jhala 2011, P. Salaris, N. Abe, and J. Laumond 2017, Ribeiro and Paiva 2014).

However, the aforementioned research is limited to either generating basic movement sequences (e.g. running, jumping) or tuning of animation to improve the believability and expression. Few works focus on the generation of novel gestures. There is a big gap in the literature in terms of improving the variety in a character's action repertoire to enhance the autonomy of humanoid characters in games (Yildirim and Stene 2008). Applying research on creativity and/or theories from creative domains is one approach to improving the creativity of procedurally generated content (Togelius et al. 2013). This is the research direction we explore in this paper via the application of Laban's Movement Analysis to procedural animation.

Some previous works have looked at frameworks like Viewpoints (Jacob et.al 2013) for analyzing human movements computationally. LMA, too, has been used extensively but most of the works have focused exclusively on Laban's Efforts (i.e. qualitative parameters of movement like *slow* or heavy) (e.g.: Dias 2007, Mentis and Johansson 2013). The *Space* component of LMA--which focuses on how people orient themselves and interact with the space--has been extensively analyzed in the performing arts (Laban 1970, Burton et al. 2016, Forsythe 2004). Other disciplines have explored how *Space* can be applied in various areas like movement inspiration (Bertol 2016), architecture (Vroman et al. 2012), and pedagogy (Block 1998). However, *Space* remains largely unexplored for computational movement analysis and generation. (Sutil 2013) argues that by viewing movements in terms of geometry, discrete units of space can be rearranged to illuminate semantics. It is to this end that we propose a theoretical framework that will allow computers to reason about human movements in terms of spatial interactions and generate creative movements using improvisation techniques.

Laban's Movement Analysis

Rudolf Laban was a Hungarian dancer and choreographer who developed a popular framework for analyzing human movements called Laban's Movement Analysis. His works, later extended by practitioners like Lisa Ullmann, Irmgard Bartenieff, and others, present an in-depth analysis of the four paradigms that make up a movement. *Body* refers to the agent that carries the movement out, *Space* refers to the surrounding environment, and the body's interaction with it. *Effort* represents the intrinsic quality of a movement (sudden, sustained, light, heavy, etc.) and *Shape* refers to the resulting form that the movement takes as a result of the interaction of *Body* in *Space* while embodying *Efforts*.

The LMA framework, while originating from classical dance studies, is broad enough to accommodate all human movements--even the ones we consider very task-specific (e.g. extending a hand to help someone). It focuses on both extrinsic (e.g. shape, spatial interactions) and intrinsic (e.g. heaviness, suddenness, flow) qualities of movements, which collectively communicate the underlying psyche of the performer. The structured, well-defined movement components make LMA well-suited for translation of complex human movements into constructs that can be leveraged by an AI system (explained further in section 1).

Laban's Space Component

A lot of research has gone into reifying LMA's *Efforts* into mathematical and physiological measures (Dias 2007, Mentis and Johansson 2013, Fdili Alaoui et al. 2017). *Space*, on the other hand, remains largely unexplored from the perspective of computational adaptation. Laban describes three kinds of spaces. *General Space* refers to the space available for movement. It could be a stage area, or game scene. *Personal Space* refers to the space that can be immediately reached by extending one's limbs. Laban defines this as the *kinesphere* (the sphere of reachability). *Interpersonal Space* is the dynamic space that results from interaction between different people in a given space.

Laban observed that classical ballet trains dancers to explore limb movements mostly in dominant directions (front, sides and back) (Clark and Ando 2014). He introduced alternative spatial structures to help the dancers explore other possibilities. These spatial structures include five platonic solids shapes--cube, tetrahedron, octahedron, icosahedron, and dodecahedron. Directing movement towards the vertices of these shapes can help dancers create dynamic, decentralized gestures (Clark and Ando 2014).

Out of the five platonic solid shapes, the *icosahedron* accommodates the proportions of the human body most accurately (Bertol 2016). The close correspondence between the icosahedron angles and the maximum angles through which our limbs move allows for a close mapping between its structure and the movement ability of the human body. (Bertol 2016) explains how the icosahedron acts as a reference system for a dancer's body to assume different postures by aligning the anatomy to different geometrical components. From a computational perspective too, the high cardinality of the icosahedron allows for a higher number of reference points (vertices, edges, planes etc.), which allows for a more granular movement mapping and control for procedural animation. Further, being the largest of the five solids, an icosahedron allows for larger bodily movements reaching out to the kinesphere and beyond. This contrasts with the dodecahedron, which promotes smaller, inward movements with a stable quality (Clark and Ando 2014).

Moving in Space: Laban-inspired Framework for Procedural Animation

We propose a theoretical framework design for an AI to computationally understand human movements in terms of Laban's icosahedron-shaped kinesphere. We also outline an improvisation engine that will use our icosahedron-based framework to generate novel movement sequences. We'd like to point out that this framework is a work-in-progress in terms of implementation.

Our proposed improvisation engine has three main components: the *movement analyzer*, the *gesture improviser* and the *movement synthesizer*. In this paper, we describe the first component in detail, introducing the structural design and encoding scheme of the framework. We supplement the explanation of each component with examples of how it would be applied in a fictional use case. The use case is described as follows: An animator is working with a creativity support tool (built based on the framework presented here) that helps them to both generate new ideas for animation and create characters that are capable of improvising novel gestures in response to game scenarios. The animator is currently working on creating animations for a character that is required to react to a sudden, frightening incident. The animator has designed an initial movement sequence in which the character's avatar throws his hands in the air and jumps.

1. Movement Analyzer

The *movement analyzer* captures a human's movements and projects them onto a virtual icosahedron to define the movements in terms of its geometric parameters. The input movement sequence could come from a motion capture system (if using a player's movements) or could be derived from the base animation clip (as in our case, the character throwing his hands in the air and jumping).

The input movement sequence is broken down into small individual gestures. Prior research in computational movement improvisation has used indicators such as shifts in rhythm and stillness as the basis for gesture segmentation (Mikhail et al., 2014). An alternative segmentation strategy could be based on joint groupings. In our use case, we can break down the action into two main components- the throwing of the hands in the air and jumping action of the legs. This strategy allows us to consider actions at varying degrees of granularity, even treating each limb as an individual group of joints.

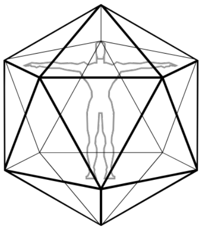

Figure 1: Icosahedron circumscribing the character

The segmented gestures are then projected onto icosahedron scaffolding that circumscribes the character's body as shown in Fig 1. This projection allows the system to define the gestures in terms of the geometric parameters of the scaffolding. We have begun to explore this aspect of the framework through the development of a Unity 3D based tool (Fig 2).

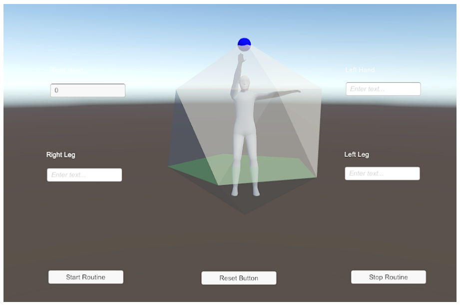

Figure 2: Unity-3D based tool for mapping a character's movements to the icosahedron.

1.1 Structure of icosahedron-scaffolding

The icosahedron shaped scaffolding is aligned with the character's body as shown in Fig 3. It is characterized by the 12 vertices, 30 edges and 30 planes of the icosahedron. These components serve as major control points in the structure, as explained in section 1.2. Apart from the vertices, any point lying on the surface of the icosahedron can be understood in terms of its relative placement to an edge or a plane. A point on an edge can be defined in terms of the ratio in which it divides the edge. A point on a plane can be defined using the perpendiculars from that point onto the edges of the plane.

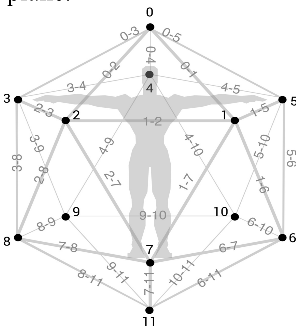

Figure 3: Icosahedron with names of vertices and edges

1.1.1 Icosahedron zones and sections

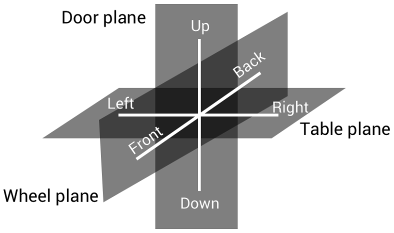

Figure 4: Dimensional cross and planes

Laban's dimensional cross defines three major directions of movement--*vertical* (up and down), *horizontal* (sideways) and *sagittal* (forward and backward) (Laban 1966). It can be seen as a 3D Cartesian space with its origin centered at the center of gravity of the character inside the icosahedron. However, a human body's frame is more rectangular rather than a single line, which expands these directions into orthogonal planes. Laban calls the horizontal the *Table* plane, the vertical the *Door* plane and the *sagittal* the *Wheel* plane (Laban 1966) (Fig 4). In our framework, we use these planes as cross-sections to divide the icosahedron into three zones, parallel to each plane. The *Table* plane divides the icosahedron into the *High*, *Medium* and *Low* zones, the *Door* plane divides the icosahedron into the *Front*, *Middle* and *Back* zones and the *Wheel* plane divides the icosahedron into *Left*, *Middle* and *Right* zones. Collectively, these zones break the icosahedron into 27 sections, as shown in Fig 5.

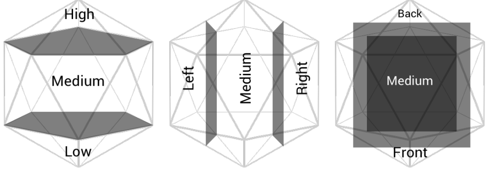

Figure 5: Different zones of icosahedron

1.2 Movement Encoding

We introduce different schemes to encode the movements in terms of the icosahedron components in this section. Each icosahedron section (as described in 1.1.1) can be annotated with a letter. The location of a limb can, thus, be defined as a combination of three letters representing *horizontal-vertical-sagittal* sections. We have *High (H), Medium(M) and Low(L)*; *Left(L), Middle(M), and Right(R)*; and *Front(F), Middle(M) and Back(B)*. The position of a joint at any point in time can be in one of the 27 sections. We introduce a term *Thick Encoding* to define this scheme of encoding gestures based on the icosahedron section. Thick encoding can be useful in identifying a general area in which a limb or a joint is located while performing a gesture. For instance, the head of the character in Fig. 3 is in the section H-M-M and if the character bends forward the head would be in the section M-M-F. In our use case we are considering, the arms of the frightened character in the final pose would be located in the H-L-M and H-R-M sections (Fig. 6).

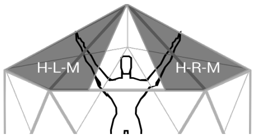

Figure 6: Arms of the character in our use case

We also introduce an alternative scheme, called *Thin encoding*, in which the location of a limb or a joint is approximated to the nearest plane, edge or vertex according to the following rules (Fig 7):

- *Vertex approximation:* When a limb is projected onto the icosahedron, it is said to point a vertex V if the projection is at a distance <= a from V. Here, the magnitude of a determines how good the approximation is.
- *Edge approximation:* When a limb is projected onto the icosahedron such that it cannot be approximated to a vertex, it is approximated to an edge E, if the perpendicular distance of the projection from the edge E is <=b.
- *Plane approximation:* When a limb is projected to an icosahedron such that it cannot be approximated to a vertex or an edge, then it is approximated to a plane P if the projection lies within the plane bounds of plane P.

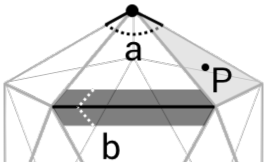

Figure 7: Thin encoding scheme for icosahedron components

In terms of thin encoding, the character in our use case, the character's arms would point to the edge 0-3 and 0-5 (Fig 3).

1.3 Human Body Restrictions

In order to ensure that the character obeys the natural restrictions of a human body, we define two concepts:

- *Unreachable components* are the components that cannot be reached by the human body from a given posture or stance. Consider the character in Fig 8. From its current stance, it cannot reach the I_5 with its right arm. It would need to twist the torso in order to reach it. This, however, depend on the flexibility of the character being animated. Our framework can accommodate differences in the bodily flexibility of characters by adjusting the definitions of unreachable components.

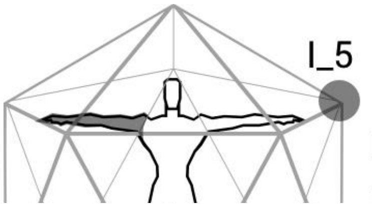

Figure 8: An unreachable component for character's right hand

- *Untraceable paths*: In some cases, a component may be reachable, but certain paths to reach it may not be viable. For example, consider a character with its right hand pointing to I_3 (as shown in Fig 9). In order to reach I_5 (a reachable component), one path that exists is I_3 → I_4 → I_5, which would require the right arm to traverse a physically anomalous path. Such non-viable paths are called *untraceable paths*. A better alternative is to travel the path I_3 → I_0 → I_5. As in the case of unreachable components, the definition of untraceable paths depends on the stance of the character, and the nature and extent of its flexibility.

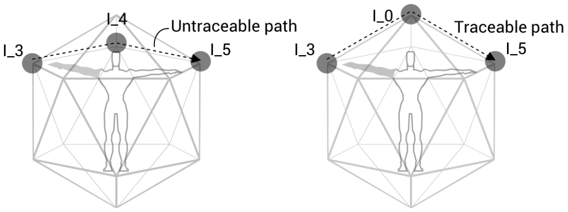

Figure 9: An untraceable path and an alternative traceable path

Adding *unreachable components* and *untraceable paths* ensures that the framework respects the natural limitations of a character's body. At the same time, if a character has an extensive movement profile, it may be difficult to define all of the untraceable components and paths for each possible stance that the character can take. One way to simplify this is to use the parent-child relationships derived from hierarchy of joints to define the natural restrictions on a joint, as well as propagate that restriction to all of the children of that joint.

2. Gesture Improviser

After translating the gesture segments into icosahedron components, the system would generate variations of these projections. The gesture improvisor can be programmed to use different transformation strategies to generate these variations. Based on our thick and thin encoding scheme, we introduce two classes of transformations. A *thick transformation* involves transformation at the section level. For example, an arm in section H-R-M can be mirrored along the Wheel plane to point to an element in the H-L-M section. A *thin transformation* involves transformations at individual component level. This provides more granular control of the limbs. If, in thin transformation, the type of input and output components are the same (e.g. both are vertices), we define it as *uniform thin transformation*; if they are different (e.g. the input is a vertex but output is an edge), we define it as *non-uniform thin transformation*.

In our use case, since the character is jumping, its torso would move from section M-M-M to H-M-M. One example of a transformation would be moving the torso from M-M-M to L-M-M instead. This could correspond to an action of ducking--as opposed to jumping--when faced with a frightening situation. Using these transformations, the system can present the animator with several possible variations of the base animation. Further research is needed into how variations in gestures can lead to a more dynamic experience for players and users.

2.1 Memorizing the improvised gestures

The appropriate movement variations that the animator chooses would then be stored by the system for future reference. The next time a player reaches the point in the game where the character throws its arms in the air and jumps, the system could select one of the variations generated by the improviser, leading to a different experience for the player. Alternatively, the improvised gestures for 'reacting to a sudden, frightening situation' can be applied to other characters in the game as well, which can further reduce animators' efforts.

3. Movement Synthesizer

The *movement synthesizer* is responsible for converting the improvised icosahedron projections back to a character's movements. It takes in the information about an icosahedron projection and how character's body should be oriented and constructs an improvised movement sequence for the animated character. After converting back, the individual gestures, the system stitches them together to form a cohesive movement response movement.

3.1 Tuning the 'style'

While our proposed framework currently focuses only on the generation of basic skeletal gesture sequences, the resulting movements could be subjected to further processing to apply certain 'style' (i.e. a set of characteristics) or personality to the movement. One can find multiple ways in the literature to achieve this (Johansen 2009, Chi et al. 2000, Bleiweiss et al. 2010, Miyake 2015, Neff and Fiume 2003, etc.). It can also be used to verify how context appropriate the generated movement is. For instance, if our system produces a variation in which the character's arms are pointing sideways to I_1 and I_2, an auxiliary system (e.g.: Castellano et. al. 2007) may point out that this variation does not quite express 'fear'. This can further assist an animator in choosing more context appropriate improvisations.

Conclusion and Future Work

In this paper, we introduced a framework that can be utilized by an AI to procedurally generate novel movement sequences for a character in 3D game environments. Our work uses Laban's *Space* component and outlines a pipeline for an improvisational AI agent that can capture human movements and project them onto a virtual icosahedron to analyze them in terms of the geometric properties of the icosahedron. The agent can manipulate the projections to generate various alternatives for which new movement sequences for the character could be generated. In the paper, we describe the structural design and the encoding schemes for the framework in detail.

There are a number of directions we are considering for future research. First, we plan on fully implementing the framework in an AI system (implementation is currently a work-in-progress) which will allow for a thorough evaluation of the quality, creativity and context appropriateness of generated gestures. Second, the framework can be extended to include the other four platonic solids from Laban. This can improve the novelty of generated gestures, just as it boosts a dancer's creative movement profile. Third, in a multi-actor scene, we can observe each individual's kinesphere to analyze the overall interpersonal engagement. This can be useful in understanding, as well as creating, more dynamic social interactions within the game. It can also be used as a tool by the digital choreographers for creating interesting choreographic patterns. Future research could also borrow the exploration of general space from (Camurri et al. 2000) for improvising movements in different parts of the 'stage' or scene.

In this paper, we take our first step towards providing a computational system with an alternative way to reason about movements--i.e. by understanding them in terms of space. This deepens the system's understanding of human behavior and allows agents to draw on improvisational dance theory to dynamically generate simulated human behavior in novel situations.

References

Antonissen, A, and Riff, M. C. 2010. An ant based approach for generating procedural animations. In *2010 22nd IEEE International Conference on Tools with Artificial Intelligence,* volume 1, 19-26. IEEE.

Bertol, D. 2016. Being Vitruvian in the Movement Infrastructure. *Symmetry: Culture and Science* (27):4.

Bleiweiss, A.; Eshar, D.; Kutliroff, G.; Lerner, A.; Oshrat, Y.; and Yanai, Y. 2010. Enhanced interactive gaming by blending full-body tracking and gesture animation. In *ACM SIGGRAPH ASIA 2010 Sketches*, 34. ACM.

Block, B. A. 1998. Keep Them in Their "Place": Applying Laban's Notion of Kinesphere and Place in Teaching Scientific Concepts. *Journal of Physical Education, Recreation & Dance* 69(3):43-47.

Brooks, R. A. 1991. Intelligence without representation. *Artificial Intelligence 47*(1): 139-159.

Camurri, A.; Hashimoto, S.; Ricchetti, M.; Ricci, A.; Suzuki, K.; Trocca, R.; and Volpe, G. 2000. Eyesweb: Toward gesture and affect recognition in interactive dance and music systems. *Computer Music Journal* 24(1):57-69.

Castellano, G.; Villalba, S. D.; and Camurri, A. 2007. Recognising human emotions from body movement and gesture dynamics. In *International Conference on Affective Computing and Intelligent Interaction*, 71-82. Springer.

Chi, D.; Costa, M.; Zhao, L.; and Badler, N. 2000. The EMOTE model for effort and shape. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques,* 173-182. ACM Press/Addison-Wesley Publishing Co.

Clark, J. O., and Ando, T. 2014. Geometry, embodied cognition and choreographic praxis. *International Journal of Performance Arts and Digital Media* 10(2):179-192.

de Souza, A. L. 2016. Laban Movement Analysis—scaffolding human movement to multiply possibilities and choices. In *Dance Notations and Robot Motion*. Springer. 283-297.

Fdili Alaoui, S.; Françoise, J.; Schiphorst, T.; Studd, K.; and Bevilacqua, F. 2017. Seeing, sensing and recognizing Laban movement qualities." In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems,* 4009-4020. ACM.

Forsythe, W. 2004. *Improvisation technologies: a tool for the analytical dance eye*. Hatje Cantz.

Hendrikx, M.; Meijer, S.; Van Der Velden, J; and Iosup, A. 2013. Procedural content generation for games: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications* (TOMM) 9(1): 1-22.

Horswill, I. D. 2009. Lightweight procedural animation with believable physical interactions. *IEEE Transactions on Computational Intelligence and AI in Games* 1(1): 39-49.

Jacob, M.; Zook, A.; and Magerko, B. 2013. Viewpoints AI: Procedurally Representing and Reasoning about Gestures. In *DiGRA conference*.

Johansen, R. S. 2009. Automated semi-procedural animation for character locomotion. *Aarhus Universitet, Institut for Informations Medievidenskab*.

Laban, R., and Ullmann, L. 1971. The mastery of movement.

Maraffi, C. T., and Jhala, A. 2011. Performatology: a procedural acting approach for interactive drama in cinematic games. In *International Conference on Interactive Digital Storytelling*, 322-325. Springer.

Mentis, H. M., and Johansson, C. 2013. Seeing movement qualities. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* 3375-3384. ACM.

Miyake, Y. 2015. Current status of applying artificial intelligence in digital games. *Handbook of Digital Games and Entertainment Technologies* 1-32.

Neff, M., and Fiume, E. 2003. Aesthetic edits for character animation. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 239-244. Eurographics Association.

Neff, M., and Fiume, E. 2004 Artistically based computer generation of expressive motion. In *AISB 2004 Convention*.

Neff, M.; Kipp, M.; Albrecht, I; and Seidel, H.-P. 2006. Gesture Modeling and Animation by Imitation.

Rett, J., and Dias, J. 2007. Computational Laban movement analysis using probability calculus. *ROBOMAT 07* 217.

Ribeiro, T., and Paiva, A. 2014. Make way for the robot animators! bringing professional animators and AI programmers together in the quest for the illusion of life in robotic characters. In *2014 AAAI Fall Symposium Series*.

Salaris, P.; Abe, N. and Laumond, J.-P. 2017. Robot Choreography: The Use of the Kinetography Laban System to Notate Robot Action and Motion. *IEEE Robotics & Automation Magazine* 24(3): 30-40.

Shapiro, A. 2011. Building a character animation system. In *International conference on motion in games*, 98-109. Springer.

Silang Maranan, D.; Fdili Alaoui, S.; Schiphorst, T.; Pasquier, P.; Subyen, P.; and Bartram, L. 2014. Designing for movement: evaluating computational models using LMA effort qualities. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* 991-1000. ACM.

Sutil, N. S. 2013 Rudolf Laban and topological movement: A videographic analysis. *Space and Culture 16*(2): 173-193.

Togelius, J.; Champandard, A. J.; Lanzi, P. L.; Mateas, M.; Paiva, A.; Preuss, M.; and Stanley, K. O. 2013. Procedural content generation: Goals, challenges and actionable steps. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Von Laban, R. 1966. Choreutics. London: Macdonald & Evans.

Vroman, L.; Naveda, L.; Leman, M.; and Thierry, L. 2012. Generating tacit knowledge through motion: A vision on the matter of space. *Art, Design & Communication in Higher Education* 10(2):255-270.

Yildirim, S., and Stene, S. B. 2008. A survey on the need and use of AI in game agents. In *Proceedings of the 2008 Spring simulation multiconference*, 124-131. Society for Computer Simulation International.