An Introduction to animation and motion blending

Stuart Bryson - 98082365

Information Technology Research Preparation

University Technology, Sydney

14/06/2005

1. Introduction 3

2. What is Animation 5

3. Motion Generation 5

3.1 Geometric Space 6

Joint Space *6*

Cartesian Space *7*

Inverse Kinematics *7*

Constraints *8*

3.2 Motion Generation Techniques 8

Key-Framed Animation *8*

Motion Capture *9*

Dynamic Simulation *10*

4. Motion Blending 11

4.1 Fundamentals of Motion Blending 12

4.2 Blending with Constraints 13

Inverse Kinematics *13*

Types of Constraints used in Blending *13*

4.3 Blending In Time 15

4.4 Blending Problem 15

5. Conclusion 16

6. References 16

Abstract

Recent years have seen an increase in the animation of articulated figures for use in films, games and robotics. This literature review aims at presenting some of the fundamentals of animation and also motion blending. Motion blending particularly suffers from dealing with differing input motions. I will discuss the techniques that have been developed to deal with this and propose a reason why all of these methods fall short.

1. Introduction

Animation of articulated figures has been a popular area of research especially in the last 20 years. Animation is used in the fields of visual effects for films, interactive video games and robotics. All of these fields, but particularly the first two, have become increasingly popular in recent years. This is due to the increasing number of films and games that rely on animated characters to add realism, perform stunts, and behave believably within their respective contexts.

This animation has been achieved through many means such as key-framed animation, motion capture and dynamics. Key-framed animation involves an animator specifying positions for the skeleton at various points in time and the computer interpolating or ‘in be-tweening” the positions of the skeleton between the keyframes. Motion capture involves a series of cameras filming a real world enactment of the motion, and usually applying this motion to a skeleton. Dynamic animation relies on a simulation of the figure using the laws of motion dynamics

In order to ensure these motions behave believably, all of these animation generation techniques can be controlled by various constraints. The constraints may be limits on joint rotations, or they may be more complex constraints such as using goals and inverse kinematic solvers.

In recent years, much attention has been paid to motion blending to achieve a wider range of animations, with this being one of their primary goals of a number of recent papers (Kovar & Gleicher 2003, p.214; Le Callennec & Boulic 2004, p.163; Bonnafous et. al 2001, p.1). Most of the research done discusses methods of blending multiple motion inputs to produce the desired output motion. While the majority of these papers achieve interesting results, all have their limitations.

This literature review will provide a survey of recent work on these techniques, their benefits and limitations. I begin by clearly defining the different animation generation techniques. Following this I discuss the various constraints used in motion generation and motion blending. I conclude with a comparison of the various motion blending techniques, and the areas in which motion blending requires more research.

2. What is Animation

Animation can be defined as a simulation of movement created by displaying a series of frames. Within this review however, I refer to animation in a more specific context. Specifically, it is the positioning of an articulated figure or skeleton over time in computer animation. I define an articulated figure as having “rigid limbs and joints with one, two, or three degrees of freedom” (Wiley & Hahn 1997, p.40). Further, an articulated figure skeleton is usually the basis for a complex character set-up with muscle and skin layers.

In order to perform animation of an articulated figure, it is important to develop a method of describing this motion. Much work has been done in the past to describe body movement. Notably, there is the annotated bibliography of body movement research by Davis (1972), and the seminal text by Badler (1972). Badler claims that despite all the available research at the time, there was a “total lack of agreement on how movement should be described”.

There are various techniques to describe motion. Some do so by describing the joint or local space rotations, others describe position and orientation of joints in world space, and others use a natural or behavioural language.

3. Motion Generation

To position a figure over time, various techniques exist that are called motion generation. This is the process of creating motion that is applied to an articulated figure. These generation techniques can be categorized into three distinct motion control categories: geometric, physical, and behavioural. Geometric methods define motion in terms of “coordinates, angles and other shape characteristics” (Magnenat-Thalmann and Thalmann, 1991, p.33). Physical methods use physical attributes to simulate motion, such as “mass, moments of inertia and stiffness” (Magnenat-Thalmann and Thalmann, 1991, p.33). Behavioural methods, which are often ambiguous, “describe behaviour in natural language terms with social, psychological, or physiological significance”, (Magnenat-Thalmann and Thalmann, 1991, p.33). Badler (1972) explains the limitations of natural language such that it is “subject to ambiguity and unavoidable imprecision in specifying positions, dynamics, styles and other aspects of movement”. Badler et al. (1991) define a more precise natural language method that has some advantages over other methods, however there is still work to be done which I qualify later.

While each of these categories is useful to distinguish control methods, they have little value when discussing motion blending. This is because the specifics of animating these motions, regardless of their category, will at some point use a geometric technique. In fact, they will all use a specific geometric method, as we will see later. For example, a physical simulation may see a figure get hit in the leg with a pole. The leg may collapse at the knee as the character falls over. In order to simulate this however, the knee joint will need to rotate, and hence we are now in a geometric category.

3.1 Geometric Space

As all motion control categories require geometric space to realise their motion, it is important to define the types of geometric space. There are many different geometric spaces; including the most common: joint space and Cartesian space.

Joint Space

Joint space is the geometric representation of the joints local rotation. That is, a joint’s rotation may be specified as a certain angle of rotation about the joint’s local axes. The most common way of representing rotations in joint space is by using Euler angles. Euler angles are named after Leonhard Euler (1707 – 1783). Given two coordinate systems xyz and XYZ with a common origin, we can specify the orientation of the second in terms of the first using three angles. For example, to rotate an elbow through a right angle, we may specify a rotation about the joint’s local x-axis of 90 degrees. There are many other ways of representing joint space including quaternions which allow for a more compact representation, but can be difficult to use in interactive environments.

Joint space is the basis for almost all computer animation. Regardless of the motion generation technique, all animation will eventually be represented in joint space. This is because joint space is “more suitable to represent and capture the intrinsic dynamics of motion” (Boulic & Thalmann 1992). I discuss this in more detail later.

One issue, however, with joint space is how laborious and tedious it is to animate with. For example, if an animator wishes to place a character’s hand on their hip, working in joint space would require them to rotate the shoulder, then rotate the elbow in order to place the hand in the right position. In fact, they will usually need to tweak the rotations of the shoulder and elbow a few times before it looks believable. It would be much easier if the animator could specify the goal, in world space, of the hand. Further, “a pure joint space technique is not suited to achieve a goal oriented specification” (Boulic & Thalmann 1992).

Cartesian Space

Another common space that animation is represented in is Cartesian space. Three dimensional Cartesian space defines 3 mutually orthogonal axes. Points are defined by specifying a distance along each axis from the origin.

Using this space, an animator can specify the location of a particular joint, such as a hand being placed on the hips. Further, animation or “motion expressed in Cartesian space is the basis of goal oriented motion” (Boulic & Thalmann 1992, p.1,6).

While this goal oriented positioning of joints is extremely useful for animators, the actual rotation of a joint must still occur in joint space. That is, while an animator may specify a goal for the end of a joint chain, each joint must still rotate accordingly in joint space. The most common method of mapping this world space translation of a joint into joint space rotation is inverse kinematics.

Inverse Kinematics

Calculating the rotation of joints in a chain to satisfy the world space position of two points on the chain is done using inverse kinematics or IK. This process, which has also been termed the inverse geometric model, has been defined as the “mapping of a desired position and orientation for an end effector onto a configuration of its supporting articulated structure” (Boulic & Thalmann 1992, p.4).

The application of inverse kinematics is much broader than the animation of articulated figures. Finding the desired joint rotations and orientation of end-effectors has been studied for years in

Robotics (Callennec & Boulic 2004, p.164, Boulic & Thalmann 1992, p.2) In fact, robotic texts "define IK as the process of determining joint angles from an end-effector position" (Gleicher 2001, p.109).

Constraints

Constraints are an important part of animating articulated figures. Constraints allow animators to enforce particular behaviours on a character. Callennec & Boulic define constraints as "the important features we need to preserve or enforce" (Callennec & Boulic 2004, p164). While this definition is somewhat abstract, it indicates the motivation of constraints, which is to achieve realistic motion and maintain those 'features' or characteristics that make the motion believable.

There are various types of constraints, but most are defined in geometric space. Joint space constraints are sometime taken into account too. Gleicher (2001) presents a more detailed discussion on the various constraints used in motion blending. Specifically, he claims that constraint techniques normally direct their attention to spatial or geometric constraints that enforce the characteristics of a figure at different points in time. Badler (1979) describes a constraint system where joint rotational limits are applied and collisions between limbs are calculated.

3.2 Motion Generation Techniques

There are many types of motion generation techniques, each with their own advantages and drawbacks. Some require animators to move the figure into specific positions over time; others aim at capturing live performances. The following is a discussion of the most popular motion generation techniques.

Key-Framed Animation

The process of creating a key framed animation is called keyframing. Keyframing is by far the most popular method for animating characters. Key framed animation is created through generating a set of poses over time. These poses are then interpolated using an interpolation technique such as linear or cubic spline interpolation. Some authors describe this interpolation process as a slerp algorithm (Bonnafous et al. 2001).

Key framed animation can be used in both joint space and Cartesian space. Keyframing in joint space is called forward or direct kinematics, conversely, working in Cartesian or world space is done using inverse kinematics. Further, the interpolation of joint space works by determining the rotation of a joint between key frames, whereas the interpolation of Cartesian space works by determining the translation and orientation of a joint and/or end-effector. Some authors claim that key framing of joint space "fails to design highly coordinated motions as walking or grasping" (Boulic & Thalmann 1992, p.3) and that complex motions such as these should be done in Cartesian space.

Some of the advantages of key frame animation include greater control by the animator to create expressive motion (Nebel 1999). Some authors argue that character design does not require physical realism but rather more concern should be placed on expressing or acting the motion for the particular character (Boulic & Thalmann 1992, p.6). Key frame animation is well suited for this as it does not represent a live performance or physical simulation.

While some authors argue that character design does not require physical realism, others insist that in order for animation to be believable it must pertain the aliveness, or the high frequency details, of motion that is not present from most keyframe animation systems (Witkin & Popovic 1995, p.1). Other limitations of key frame animation include the inability to create a rich variety of motion in real time (Wiley & Hahn 1997 p.39).

Motion Capture

Motion capture is a technique used to capture the real time performance of an actor or another type of real time motion input (Boulic & Thalmann 1992, p.2). There are various types of motion capture rigs or set-ups. They all share the fundamental goal of providing data from the live performance using sensors (Bonnafous et al. 2001). They do this by triangulating the position of sensors placed on the actor as they perform the motion. The positions are then mapped directly onto the joints in joint space using, among other techniques, inverse kinematics (Callennec & Boulic 2004, p.164). The main differences between the various methods are in the sensor technology. Some use magnetic sensors, others use optical, and some of the earlier methods use mechanical rigs.

Motion capture has proven to be a good technique for generating realistic human animation (Kovar & Gleicher 2003, p.214, Callennec & Boulic 2004, p.163). Some authors state that motion capture not only produces highly realistic character motion, but that it does so much more efficiently and with little effort when compared with the more traditional methods (Witkin & Popovic 1995, p.1).

Other advantages of motion capture include the ability to produce motion more rapidly (Wiley & Hahn 1997, p.39, Callennec & Boulic 2004, p.163). It can be used to create custom animations, and it can be used to create a library of reusable animation clips (Witkin & Popovic 1995, p.1). For this reason, it is often used for film and games (Wang & Bodenheimer 2004).

With so many advantages of motion capture come just as many disadvantages. Many authors state the lack of flexibility of motion capture animation (Wang & Bodenheimer 2004). There is an innate lack of control with captured data (Wiley & Hahn 1997, p.39). Not only that, but as motion capture is done in real time, any animations that are required must be thought of, and planned out, before the capture is done (Callennec & Boulic 2004, p.163). Hence, if a particular motion is missed, it is not always possible to capture it at a later stage in the motion capture rig. If it is possible it is usually at great expense.

Boulic and Thalmann (1992) raise some further disadvantages with motion capture. They discuss the technical limitations of current motion capture technology. They claim that currently the sensors placed on an actor do not accurately correspond to the actor's joints, and secondly that the calculation of the sensors location is proportionally related to the field of view of the cameras and the quality of the image analysis (Boulic & Thalmann 1992 p.2-3).

There are many works dedicated to the problem of editing motion capture data, which by nature is hard to work with (Boulic & Thalmann 1992, Gleicher 2001). There is still much to be researched in this field, especially as motion capture becomes more popular.

Dynamic Simulation

The last of the three most common motion generators is the dynamic or physical simulation. Physical simulations use physical properties and forces to determine joint space rotations. It is the only motion generation technique that uses physical laws (Bonnafous et al. 2001). The simulations usually involve Newton's laws and specify the torque on each joint (Boulic & Thalmann 1992, p.3).

While dynamic simulation offers real promise (Wiley & Hahn 1997, p.39) as it is able to model realistic and believable motion without the need for any motion capture, it still has significant drawbacks. These include "lack of control, difficulty of use, instabilities and computational cost that usually precludes real-time operation" (Wiley & Hahn 1997, p.39). Other authors have stated the limitation in real time environments (Bonnafous et al. 2001). One last limitation of dynamic simulations is the difficulty of determining the resulting motion of a simulation before it occurs (Boulic & Thalmann 1992, p.3). Some applications may require an approximation of the resulting motion in order to use it.

For these reasons, dynamic simulation is rarely used in animating articulated figures. If it is used in a real-time environment such as a video game, it will be a fairly simplistic simulation; although as computing power increases, these simulations are becoming more complex. It is also possible to use a dynamic simulation that is not real-time to generate a key framed or 'baked' animation of the simulation for reuse in a real-time situation. This is similar to reusing motion capture animations.

4. Motion Blending

With a good understanding of animation of articulated figures and motion generation techniques, it is possible to discuss the concept of motion blending. Motion blending is used for many reasons. Transitions between two motions are commonly used in interactive applications such as video games. Some authors claim that "motion blending is commonly thought of as creating the transition of an animated figure to the first frame of a piece of motion" (Polichroniadis & Dodgson 1999, p225). While transitions are a common application of motion blending, there are many others. A further application involves generating a motion using the interpolation of an arbitrary number of input motions to "produce a parameterised space of motions" (Kovar & Gleicher 2003, p.214). Blending can also be used to take advantage of the best attributes of more than one motion generation technique. For example, a particular application may require the subtleties of motion capture but the flexibility of key framed animation. By using motion blending, the best of both worlds can be achieved.

It is important to note the difference between motion blending and motion editing. While both have common concepts and technical difficulties, there exists a fundamental difference between the two ideas. Motion editing aims to alter or adapt only one animation (Menache 1999) whereas motion blending aims to blend more than one animation and in many cases will consider an entire motions from a database to generate new motions (Callennec & Boulic 2004).

One author claims that motion blending is thought of as "creating the transition of an animated figure to the first frame of a piece of motion" (Polichroniadis & Dodgson 1999). This definition is quite limited. First, as already mentioned, motion blending is not limited to transitions. In fact some authors sole focus is on expanding the range of motions, and not necessarily transitioning between them (Wiley & Hahn 1997, Kovar & Gleicher 2003). Second, when blending is used for the purpose of transitioning between motions, it is not necessarily limited to the first frame of the motion.

4.1 Fundamentals of Motion Blending

Motion blending is achieved by taking an arbitrary number of input motions and, based on particular requirements of the blending technique, determining the output motion. Most techniques apply the blending on a per joint basis. That is, for each joint in a skeletal hierarchy, the blend for that joint is calculated with no attention being paid to other joints in the hierarchy.

When the technique works on a per joint basis, not all joints require blending. If a joint is being driven by only one input motion then it simply receives that inputs transformation. If there are multiple input motions driving a joint, then that joint's transformation must be blended. Bonnafous et al. describe it in this way:

> There are two cases: either a bone is associated to a motion generator and there is not a conflict; or a bone is associated to a set of motion generators and there is a conflict… (Bonnafous et al. 2001, p.4)

When a joint is in conflict it must be blended. The advantage of blending on a per joint basis means there are less conflicts. If for example two different motion inputs drive separate joints there is no conflict. Bonnafous et al. describe it this way. "There [is] no conflict when we use several motion generators on different bones". They describe this as cooperative motion blending (Bonnafous et al. 2001, p.4).

Blending algorithms such as the one outlined above are a simple and very fast way to achieve basic motion blending. Using such a simple algorithm however does not produce acceptable results for articulated figures. The motion is not believable. This is because the joints do not behave in a way that is consistent with the expected behaviour or characteristics of the figure. To maintain this believability, it is important to satisfy constraints on the motion.

4.2 Blending with Constraints

In order to achieve realistic motion, most modern blending techniques take constraints into account. Constrains ensure that the motion is believable and that the features or characteristics of the motion are maintained. When blending motions however, constraints may not be satisfied or sometimes ignored resulting in unbelievable motion. Motion blending must be able to deal with blending constrains while maintaining the motions believability.

Inverse Kinematics

In a motion blending context, Inverse Kinematics are required to solve spatial constraints. In fact, "all constraint-based motion editing methods must include a solver for IK problems in one form or another" (Gleicher 2001, p.110). While the author does not make this clear, their statement is only true when referring to spatial constraints. The most common way that a motion is constrained is using an IK chain and animating the end effector – be it through key framed motion generation or other techniques. For example, an IK chain may be rigged onto the leg of a skeleton such that the root of the IK chain is the hip and the end effector is placed on the ankle. As the end effector is animated, the IK solver is responsible for solving the rotations of both the hip and knee joints. With a clear definition of constraints in an animation context it is possible to discuss the difficulties of motion blending while respecting constraints.

Types of Constraints used in Blending

Constraints are one area of disparity of motion blending techniques. As discussed above, there are many techniques for solving constraints in animation. However, multiple input motions and the desire to blend between them further complicate these techniques. While most blending techniques use weighting or priorities, the specifics of how these techniques work however, differ somewhat.

The more common approach is to use a method that uses priorities atop weighting. Each constraint is given a priority. When a joint is being solved, all constraints influencing the transformation of that joint are taken into account. The joints with a higher priority are solved first and then the lower priorities are be considered. If a joint is in conflict because two constraints of the same priority are influencing it then a weighting scheme is used to blend the transformation between the constraints. Many have used a method such as this (Bonnafous et al. 2001, p.5, Callennec & Boulic 2004, p.165).

Kovar and Gleicher on the other hand use a constraint matching technique whereby a constraint is ignored if there are not corresponding constraints in all the input motions (Kovar & Gleicher 2003, p.219). For example, if a foot constrained to the floor in one input motion but not another, the constraint is ignored. While this may work in many situations, in others it may ignore important constraints when it should not. For example, lets say we have 4 input motions, 3 of which satisfy a constraint that is vital to the characteristics of their motion. The last input motion however, does not satisfy the constraint. If this last input motion is not imperative to its blended motion characteristics, the motion will not be believable as the constraint is missing.

Normally, when solving constraints, they are solved within their respective IK chain. Callennec & Boulic, however, use their own IK solver to specifically create a more flexible constraint system. They propose a more general technique whereby a constraint is allowed to “recruit all or part of the joints from its parent joint up to the root” (Callennec & Boulic 2004, p.165). This enables a method whereby only the joints required to satisfy the constraint are included in the solver. Further, not only is computation time decreased as there are fewer joints being considered in the chain, but more importantly this allows more realistic results.

Currently, none of the aforementioned techniques are completely successful at seamlessly, and with little effort, blending multiple input motions while maintaining believable human motion. Kovar and Gleicher go a significant way in solving this, however it still requires certain assumptions to be made on the input motions.

4.3 Blending In Time

Another important consideration in motion blending is time. Regardless of the application, in order for motion blending to work correctly the input motions must be aligned temporally to achieve believable results. In order to create a seamless blend, "a good transition point between motion clips" must be determined (Wang & Bodenheimer 2004, p.337).

Linear blending is the most popular method as it is "simple and often generates visually pleasing results" (Wang & Bodenheimer 2004, p.339). Many linear blending methods do not consider temporal issues. They make the assumption that the first frame is the start of a motion. This results in a very limited range of transitions achievable but more importantly requires the input motions to meet a specific range of requirements in order for the motions to be blended.

Polichroniadis & Dodgson propose a classification method to match frames in different input motions. They do this by "removing the assumption that each piece of motion must start at the first frame" (Polichroniadis & Dodgson 1999, p.225). While this method adds significant flexibility it only helps match one frame in the transition. If during the transition there are two important frames that should be matched their system will not work.

Kovar and Gleicher (2003) solve this problem of timing using Coordinate Frame Alignment and Timewarping. Like the previous method, they determine one frame in multiple input motions as being similar. This they call the coordinate frame. Using a timewarp curve the input motions timing is then adjusted such that each motion follows similar timing and can thus be blended more efficiently.

While frame matching does increase the validity of the blended motion, it is clear that this on its own does not allow blending of arbitrary motions. There are still significant requirements placed on the input motions to be of similar motions and or timings. Using frame matching along with timewarping furthers the chance of the blended motion looking believable.

4.4 Blending Problem

I speculate that the main problem with motion blending is the level that this blending occurs. The methods presented above perform the motion blending at the joint space level. Consequently, the blending algorithms must take measures to compensate for differences in motion input parameters. I further speculate that motion blending may perform better when under less strict situations with fewer requirements on the input motions. Therefore, if there existed a method of describing motion at a higher level, like the natural language presented by Badler et al. (1991), that was very precise and unambiguous in meaning, then motion blending would no longer suffer from the exertion it currently does.

5. Conclusion

I have presented an introduction into the field of animation and motion blending. I have discussed motion generation and the techniques that animators use to generate motion. I have also discussed the various methods for motion blending in use today. While recent years have produced new and exciting techniques to blend motion while retaining believability, there still remains the need for further work. I have proposed that perhaps the main problem with motion blending is the level at which the blending occurs. Instead of focusing the blending algorithms in the joint space levels, with a less ambiguous but more general description of motion, motion blending may become more believable.

6. References

Badler, N.I., O'Rourke, J. & Kaufman, B. 1980, 'Special Problems In Human Movement Simulation', *ACM*, pp. 189-197.

Badler, N.I. & Smoliar, S.W. 1979, 'Digital Representations of Human Movement', *Computing Surveys*, vol. 11, no. 1, pp. 19-38.

Bonnafous, V., Menou, E., Jessel, J.-P. & Caubet, R. 2001, 'Co-operative and Concurrent Blending Motion Generators', *WSCG*, ed. V. Skala.

Boulic, R. & Thalmann, D. 1992, 'Combined Direct and Inverse Kinematic Control for Articulated Figure Motion Editing', *Computer Graphics Forum*, vol. 11, no. 4, pp. 189-202.

Callennec, B.L. & Boulic, R. 2004, 'Interactive Motion Deformation with Prioritized Constraints', *Eurographics/SIGGRAPH Symposium on Computer Animation*, Eurographics Association, Grenoble, France, pp. 163-171.

Gleicher, M. 2001, 'Comparing Constraint-Based Motion Editing Methods', *Graphical*

Models, vol. 63, no. 2, pp. 107-134.

Jung, M.R., Kalita, J.K., Badler, N.I. & Ching, W. 1991, 'Simulating Human Tasks Using Simple Natural Language Instructions', *Winter Simulation Conference*, eds C.L. Nelson, W.D. Kelton & G.M. Clark.

Kovar, L. & Gleicher, M. 2003, 'Flexible Automatic Motion Blending with Registration Curves', *Eurographics/SIGGRAPH Symposium on Computer Animation*, Eurographics Association, San Diego, California, pp. 214-224.

Magnenat-Thalmann, N. & Thalmann, D. 1991, 'Complex Models for Animating Synthetic Actors', *Computer Graphics & Applications*, vol. 11, no. 5, pp. 32-44.

Menache, A. 1999, *Understanding Motion Capture for Computer Animation and Video Games*, 1st edn, Morgan Kaufmann Publishers Inc.

Nebel, J.-C. 1999, 'Keyframe Animation of Articulated Figures Using Autocollision-Free Interpolation', *17th Eurographics UK Conference'99*, Cambridge, UK.

Polichroniadis, T. & Dodgson, N. 1999, 'Motion Blending Using a Classifier System', *WSCG (Seventh International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media '99)*, University of West Bohemia, Plzen, Czech Republic, pp. 225-232.

Wang, J. & Bodenheimer, B. 2004, 'Computing the Duration of Motion Transitions: An Empirical Approach', *SIGGRAPH Symposium on Computer Animation*, eds R. Boulic & D.K. Pai, Eurographics/ACM.

Wiley, D.J. & Hahn, J.K. 1997, 'Interpolation Synthesis of Articulated Figure Motion', *IEEE Computer Graphics and Applications*, vol. 17, no. 6, pp. 39-45.

Witkin, A. & Popovic, Z. 1995, 'Motion Warping', *SIGGRAPH Computer Graphics Proceedings*, ACM, Los Angeles, pp. 105-108.