Real-Time Expressive Gaze Animation for Virtual Humans

Marcus Thiebaux
USC/ISI
4676 Admiralty Way, Suite 1001
Marina Del Rey, CA 90292
thiebaux@isi.edu

Brent Lance
USC/ICT
13274 Fiji Way
Marina Del Rey, CA 90292
blance@ict.usc.edu

Stacy Marsella
USC/ICT
13274 Fiji Way
Marina Del Rey, CA 90292
marsella@ict.usc.edu

ABSTRACT

Gaze is an extremely important aspect of human face to face interaction. Over the course of an interaction, a single individual's gaze can perform many different functions, such as regulating communication, expressing emotion, and attending to task performance. When gaze shifts occur, where they are directed, and how they are performed all provide critical information to an observer of the gaze shift. The goal of this work is to allow virtual humans to mimic the gaze capabilities of humans in face to face interaction. This paper introduces the SmartBody Gaze Controller (SBGC), a highly versatile framework for realizing various manners of gaze through a rich set of input parameters. Using these parameters, the SBCG controls aspects of movement such as velocity, postural bias, and the selection of joints committed to a particular gaze task. We provide a preliminary implementation that demonstrates how related work on the Expressive Gaze Model (EGM) can be used to inform management of these input parameters. The EGM is a model for manipulating the style of gaze shifts for the purpose of expressing emotion [11]. The SBGC is fully compatible with all aspects of the SmartBody system [23].

Categories and Subject Descriptors

I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism – *Animation, Virtual Reality*

Keywords

Virtual Human, Procedural Animation, Gaze

1. INTRODUCTION

Gaze is a key aspect of face to face human interaction. Over the course of an interaction, an individual's gaze can perform a variety of functions. For example, a gaze could be used to attend to task performance [2], regulate conversational interaction [8], express intimacy or emotion [2], or exercise social control [3]. In order to send this wide variety of signals through gaze, humans use a wide variety of different gaze manners. The properties of these gaze manners can include when a gaze shift occurs, where it is directed, and how it is performed.

The current state of the art virtual humans lack equivalent capabilities for communication and expression through gaze that humans display in face to face interaction. In this paper we seek to address this by introducing the SmartBody Gaze Controller (SBGC), which is a procedural animation system that can produce full-torso gaze shifts (or gaze aversions) towards any arbitrary target, and provides broad capabilities for the realization of a wide variety of gaze manner in virtual humans. One goal of the SBGC is to support research into gaze models for virtual humans. In this way, we hope to assist the development of virtual humans that can use gaze in the same ways that humans do in face to face interaction.

The SBGC is designed to work within SmartBody, an open-source and highly versatile system that can produce interactive animated gaze shifts using a skeleton-independent API. Because the SBGC is implemented as a controller within the SmartBody system, it benefits from the full capabilities of SmartBody [23]. Specifically, the SBGC can be automatically scheduled and blended with other animation controllers that implement other aspects of verbal and nonverbal communication, or that use animation generated via keyframing or motion capture.

The SBGC can also produce a wide variety of gaze shifts in various manners. The flexibility of the SBGC is provided through a rich set of input parameters that control aspects of movement such as velocity, postural bias, and the selection of joints committed to a particular gaze task.

However, the very richness of this space of parameters raises the question of how to set their values; in other words, how to map a particular gaze model to the space of parameters. To demonstrate this, we provide a preliminary implementation of a mapping from the Expressive Gaze Model (EGM), a model for manipulating the style of gaze shifts for the purpose of expressing emotion [11], to the SBGC input parameters. This will also demonstrate the versatility of the SBGC by showing how the input parameters of the SBGC can be used to produce the wide variety of gaze styles required to implement the EGM.

In this paper we will discuss the SmartBody Gaze Controller, its input parameter space, and provide a preliminary implementation of a mapping of the Expressive Gaze Model to the SBGC's parameter space.

2. RELATED WORK

As previously mentioned, there are many different functions of a single individual's gaze. And there are many different virtual human gaze models that plan and realize gaze behavior in order to produce realistic or communicative gaze behavior. For example, there are gaze models driven by the internal state of the agents, such as the Rickel gaze model's integration with task planning or execution [12, 19], as well as gaze models that perform

Cite as: Real-Time Expressive Gaze Animation for Virtual Humans, Marcus Thiebaux, Brent Lance, Stacy Marsella, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 321–328

impression management, attempting to express emotion or influence the user's internal state, such as [4, 5, 18].

In addition to modeling gaze, the methods used to produce animation for both the SBGC and the GWT are based in the field of human figure animation. For example, the full spine model described by Monheit and Badler [16] provides capabilities similar to those provided by SmartBody's spine control system. Real-time inverse kinematics solvers have also been previously addressed [20]. There are also several general methods for producing expressive animation, such as [1, 17]. However, to date these techniques have not been combined into an open-source procedural framework for expressive gaze that provides the power and capabilities of the SBGC.

The SBGC is a procedural controller capable of realizing gaze shifts to arbitrary stationary and moving targets using a highly flexible and expressive framework for controlling the physical manner of those gazes. It is also designed to be combined with other simultaneous controllers within SmartBody. This allows a much broader spectrum of gaze behaviors to be realized, as well as providing a basis for future implementations of gaze behavior in virtual humans.

The SmartBody Gaze Controller has been used in a large number of systems, including Stability and Support Operations Simulation and Training (SASO-ST) [22] a platform for virtual human research, and ELECT BiLat [6], a fielded training system designed to teach cross-cultural negotiation skills that was developed by an external game company. No modifications to the SBGC have been necessary for any of these projects.

3. GAZE CONTROL IN SMARTBODY

The SmartBody Gaze Controller provides a highly flexible framework to enable rapid and reusable research into gaze behavior for virtual humans. The SBGC can realize gaze shifts to a target by applying motion calculations to a selected set of skeletal joints. We refer to the contribution of each joint involved in this shift as a gaze-joint *task*.

Once provided a target, the SBGC allows many different ways to realize a gaze shift directed at that target. One way it does this is by utilizing different selections of gaze-related joints. For example, a gaze shift can be performed with just the eyes, or just the head, or with the entire head and torso. In fact, multiple gaze controllers can be layered simultaneously, so that an underlying gazing task can be modified by a quick glance.

The SBGC also provides the ability to specify both targeting offsets, and directional biases for each joint involved in the gaze shift. Offsets can be used to perform gaze aversion, by applying an angular offset in the target's direction. Modulating expressive behavior, such as bowing the posture during a gaze, can be achieved by applying vertical directional biases to the joints of the back or neck.

Further, the SBGC is highly responsive. It is self animating, and actively seeks moving targets. It automatically realizes eye convergence and the Vestibulo-Ocular Reflex (VOR) by separately animating each eye to focus on the target independent of head motion. The SBGC provides full head and torso control for the realization of gaze shifts with respect to a target for the express purpose of indicating a character's visual and communicative attention.

The various gaze parameters allow for variation in the expressive manner of attention, from aversive postures to postural biases, and from casual glances to intense focus. Parameters allow adjustment of smoothness and abruptness in gaze shifts and target tracking, and include blending weights that allow underlying postural layers to contribute to and modify the resulting motion (Table 1). Access to these parameters is provided by the Behavior Markup Language or BML commands [24] to SmartBody, as described in the SBM gaze documentation [21].

Table 1: Gaze Joint Parameters

Parameter	Description
Task Weight	Contribution toward objective
Joint Selection	Which joints are used
Priority Joint	Which joint to reach objective
Joint Limits	Range of joint movement
Joint Speed	Maximum instantaneous joint speed
Task Time	Approximate time to objective
Smoothing	Weight of decaying average function
Blend Weight	Strength of blend with underlying pose
Joint Bias	Postural predisposition
Targeting offset	Aversion relative to objective

3.1 Gaze Motion Synthesis

Gaze motion in the SBGC involves generating angles for a kinematic chain of joints that assist an end effector, typically the eyes, in aligning their forward axes with a target. Our solution involves semi-independent, parameterized joint operators, combined with a heuristic direct search technique, known as *cyclic coordinate descent* (CCD) as outlined in [17], which has been used extensively to efficiently resolve narrowly defined problem spaces in inverse kinematic robot manipulation [7]. Our approach leverages the iteratively converging feature of CCD, emphasizing computational efficiency, as well as parameterized expressivity.

Most joints involved in gaze-target acquisition are typically engaged in assisting the eyes in acquiring their target, not in acquiring the target themselves. As a result, a proper solution is more complex than reorienting each joint to align its coordinate axis to the target. It must also take into account the translational offset from each joint origin to a point between the eyes. Therefore, the alignment algorithm is not just a vector direction calculation. It involves the alignment of a *ray* (a direction combined with translational offset) toward the target. Each joint has a fixed translation and variable orientation relative to its parent (shown in Figure 1). The joint-forward ray represents the gaze end-effector, typically the eye and its sight-line, relative to each joint.

In our control model, a gazing action is not simply a pose onto which the joints are made to converge through blending or interpolation. The future position of a moving (or substituted) target is not known, and the immediate task simply involves moving the gaze incrementally toward the current target position. The action is resolved on an iterative frame-by-frame basis, by assigning a task to each joint, and applying a set of prescribed constraints.

The SBGC coordinates the tracking actions of individual joint motors [19] that function independently to rotate within the coordinate frame of each joint's hierarchical parent. The SBGC

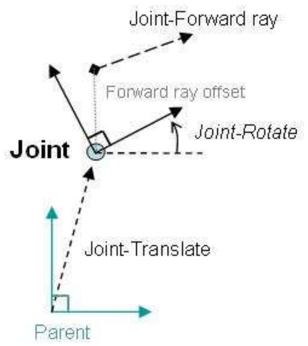

Figure 1: Joint Details Showing Forward Ray

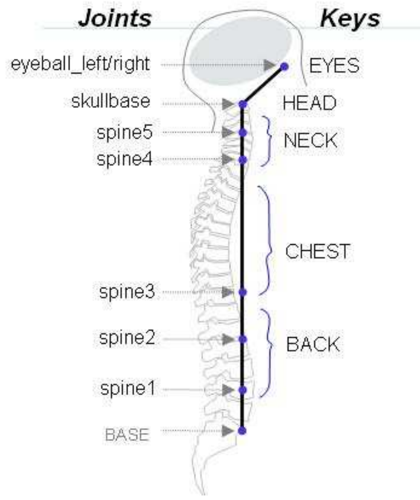

Figure 2: Keys Mapped to Joints

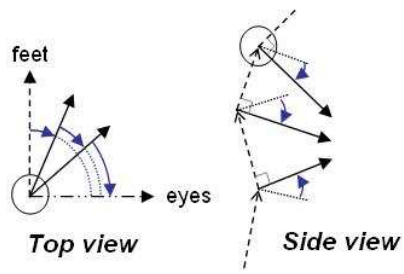

Figure 3: Task Weight is Not a Sweep Division

assigns each joint a task, which entails making up some portion of the entire required gaze shift relative to its parent joint, regardless of the parent's configuration and disposition.

A single *descent* of the joint hierachy makes small corrections from the current state towards a converging state with each animation frame. It would be a simple matter to control the number of convergence iterations per second by repeating iterations at lower frame rates, in order to guarantee a convergence rate, because the per-frame calculations are lightweight.

Motion is induced by moving each joint toward its assigned goal, while applying speed and smoothing constraints that prevent the joint from moving immediately to that goal. As joints reach their individual prescribed range-of-motion limits, their movement is clamped, and what remains of the overall shift is left up to other joints that can still move. Since each joint attempts to fulfill a shortest-path rotation toward its target, if the target moves behind and around the virtual body, each joint will automatically switch directions, and together the joints will turn the spine and head around toward the new target position without external intervention. Speed limits, joint limits, and all other active parameters are applied continuously to produce smooth and persistent target tracking.

3.2 Gaze Manipulation

Rather than exposing the underlying skeletal decomposition of the character to the client API, joint parameters are manipulated through an abstraction of standardized gaze keys, which serve as *logical joints*, as shown in Figure 2. For example, to control the limits of neck motion, the client associates a limit value with a neck key. This value is then internally distributed between the two neck joints spine4 and spine5, so that the limit of the neck joints as a group match the desired limit for the corresponding key. Conversely, a blending weight, which determines the contribution of an underlying pose, is calculated as an average of the weights from nearby keys.

Keys are used to manipulate the underlying skeletal joints without requiring explicit knowledge of the skeleton configuration. Conceptually, a key refers to a bodily segment spanned by joints rather than a jointed position in the body. Gaze keys allow a common API to control skeletons with a different number of joints in the spine, which may be increased for more realistic and smooth spine behavior. It is a straightforward matter to generate key-joint parameter mappings dynamically for both distribution and averaging based on joint spacing.

3.2.1 Task Weight

The set of joints committed to a gaze shift must be coordinated in their objective. This is initiated by calculating the relative contribution of each joint toward the gaze shift objective, usually eye-to-target alignment, and assigning this value to each joint as a task weight. This task weight determines how much of the required overall shift each joint must fulfill. Seen from above, this looks like a simple division of the sweep angle toward the objective, but from the side it is clear that not all joints will sweep in the same direction, as shown in Figure 3. Therefore this is not simply a matter of dividing up a sweep angle among the joints, but rather as a weighting of contributions relative to each joint parent. The contribution each joint makes toward the objective is calculated relative to the parent of each joint, and depends on its location in the joint chain from the eye, rather than the overall sweep from the body coordinate frame.

If the task of a gaze is to align the eyes to a target, the task-weight of the eyes is set to 1, and the task-weight of the head joint is set to 1/2 so that the head attempts to make up half the angular distance from the neck angle to the desired eye angle. The neck joints acquire a task-weight of 1/3, 1/4, and so on. With 7 joints from spine to eye, the bottom spine joint has a task-weight of 1/7, meaning that it will try to move 1/7 of the angular distance to the target.

This automatic weighting produces sufficiently compelling results that they are not typically manipulated. In order to facilitate experimentation with the task-weights, we consider modifying them with an input array of normalized values. However, direct control over these internal task-weights is not intuitive, and can produce results that are difficult to interpret.

3.2.2 Joint Selection and Priority

In SmartBody, gaze controllers can be layered so that an underlying gazing task involving the full spine and eyes can be modified by a quick glance or deictic gesture involving the head.

The SBGC can be set to operate over any contiguous subset of joints in the set. A gaze action may be chosen to involve only the eyes and neck, or could involve the full spine. Some useful gaze-like gestures can be generated by movement of the chest alone toward another direction, in conjunction with a full torso gaze.

A priority-joint value determines which joint is expected to actually acquire the target. Typical conversational behavior often entails alignment of not just the eyes, but also the whole head toward the counterparty. This configuration maintains conversational focus, while allowing the eyes to move elsewhere from moment to moment according to other simultaneous communication requirements. The priority joint is used as a basis to generate the task-weight values.

As the priority joint is moved downward along the spine, the lower joints acquire a heavier weighting by this formula than they would by default, and the degree of bodily commitment, and hence visible intensity, of the gaze shift increases. For example, if the bottom neck joint is chosen as the priority, the joints in the neck and head acquire a task-weight of 1, and joints below the priority joint are assigned weights of 1/2, 1/3, etc.

3.2.3 Joint Angle Limits

Bodily joints have limited ranges of motion. The forward axis of each joint in SBGC is allowed to swing freely within an ellipse transcribed onto the sphere of possible swing values. This is a common and computationally efficient way of describing the limits of motion for all joints. The heading component of joint swing is symmetric, whereas the pitch component allows different ranges for pitching up as for pitching down. Thus, 4 values are required to limit joint range: pitch-up, pitch-down, heading-magnitude, and twist-magnitude.

Our limits are simply hard clamps on the swing and twist values of the joint orientation, and will induce highly visible accelerations in joint movement in the absence of smoothing methods. We are examining the use of non-linear elastic limits that respond to the magnitude of a targeting task, which will resolve these discontinuities in velocity, but may also further complicate the prediction of the time required to complete a gaze shift.

3.2.4 Joint Speed and Task Time

The joint rotation speed of a gaze motion is specified in terms of how fast the head will be allowed to move with respect to the body frame, and a separate eye motion speed. The body-head speed is used internally to control individual speeds of joints involved in the gazing task by dividing this overall speed according to their proportional task weights. Individual joint speeds serve as a clamp on the maximum speed that a joint can rotate during a given animation frame. Since the net actual speed of the head depends on the coordinated motion of all gaze joints together, and since joints may quickly come up against their motion limits, actual measured speed is usually markedly less than this speed parameter setting.

The use of proportional task weights to distribute overall speed was chosen based on observation of the resulting animations, and is subject to further study. The coordination of multiple speeds for multiple joints is an unwieldy task if left up entirely to the client. Inappropriate selection will result in some joints fulfilling limits too early while other joints continue to move, producing a plainly unnatural appearance of overall motion.

We observe that the apparent intensity of gaze-target acquisition can be discerned by the amount of time elapsed between motion initiation and acquisition. Sudden gaze shifts appear more intense than gradual shifts. In particular, this apparent intensity is largely independent of task magnitude, meaning that it is the total elapsed time that is important, not the specific rotational speed that is achieved. Since our implementation involves loosely coordinated joints with individual speed settings, we supply a time-hint setting that attempts to complete a gaze shift within the time provided, in order to produce a desired level of apparent intensity.

3.2.5 Joint Smoothing

When joints are moving, sudden accelerations are produced during initial controller activation, as well as the moment when a joint arrives at its limit. Smoothing is achieved with the use of a decaying average function, which blends the desired next angle with its previous, to produce a current value. This is a computationally efficient way to reduce the robotic appearance of joints constrained by speed alone. The smoothing adjustment is applied after speed and joint angle limits are applied. The output of these constraints supply the next desired angle for the smoothing algorithm.

Aside from computational simplicity, the key advantage of a dynamic averaging method over the explicit construction of smooth spline paths is that it does not require a known future value, and so is readily suitable to sudden interruptions in motion. Under normal conditions, this function will produce a gradual acceleration up to the joint speed limit, and a gradual deceleration as it arrives at the target, with a minor, tunable latency. Joints such as the eyeball will generally have little or no smoothing and high speed, simulating the eye's ballistic movement, while the lower spine will have a high smoothing value.

3.2.6 Blend Weight

A gaze controller instance operates within a layered hierarchical framework of other controllers of various types, such as full body motions and poses, as well as those for nodding, head-shakes and other instances of gaze. In order to avoid completely overwriting underlying spinal poses, each gaze joint maintains a separate blending weight which is used to linearly combine its desired angle with the underlying angle. For example, we would expect a walking motion that involves some torso twisting to continue contributing some spinal motion to the final output even when a gaze task is modifying it. In practice, lower back joints may incorporate more of the underlying pose, while the eyes would overwrite completely, since they are the primary gaze end-effectors.

3.2.7 Joint Bias

We use the term *bias* to capture the idea of a joint's postural predisposition toward a target. While each joint attempts to assist the eyes and head in acquiring their target, we can achieve subtle communicative nuances in the manner of gazing motion by altering the conceptual notion of 'forward' for any given joint. For example, a character may maintain eye contact with a counterparty while simultaneously turning the face slightly to the side. This modification may be used to convey a query or sense of confusion. A similar adjustment may be made to face down to convey humbleness, or up for pride.

We view this adjustment as conceptually distinct from aversion, described in the following sub-section. Gaze aversion modifies the gaze target direction rather than the definition of what is forward for a given bodily segment. The bias of the head determines what is being shown in the direction of the counterparty, and can be specified differently for each part of the spine and head for a variety of effects. A character that rolls the eyes up in disdain is not so much looking away from the audience. Rather, the character is showing the whites of the eyes to the audience. Other related social dynamic cues can be communicated with parts of the torso, such as despair or dominance.

3.2.8 Targeting and Aversion

The target for a gaze task can be specified in terms of a point in space, a direction in the world, or most commonly in terms of a named joint in the scene graph that supplies a dynamically changing position. When the target is a named joint, its position is resolved during each animation frame to generate a point target in world coordinates. As the gaze controller is evaluated, this point is transformed to the coordinate frame of each joint's parent, and the gaze-joint task is resolved as the required rotation of that joint, in its parent's frame, to achieve forward-ray alignment, subject to weights and constraints outlined above. An optional *object offset* vector in the target joint coordinate system can be specified in order to localize some part of the object's spatial geometry that is represented in the scene by the target joint.

The controller API also provides for a *targeting offset* that is distinguished from the object offset, and a separate coordinate frame for that targeting offset. This secondary offset is used to realize gaze aversion, which typically means an angular deviation from the direct line of sight to the target. This deviation is specified in the coordinate frame of the character's own body, not of the target, and the API allows for different joints to be selected for this purpose to achieve different effects in the final posture.

4. EXPRESSIVE GAZE IN SMARTBODY

The SmartBody Gaze Controller has an extensive set of parameters that allow the generation of a large variety of gaze styles. In order to provide inputs to the SBGC, and allow a simpler set of high level inputs, we provide a preliminary implementation of a mapping between the Expressive Gaze Model (EGM) and the SBGC input parameterization. The EGM is a model for manipulating the style of animated gaze shifts for the purpose of displaying emotion. It currently runs as an offline animation generation system. The EGM comprises two models of movement control:

- A procedural model of eye movement based on stereotypical eye movements described in the visual neuroscience literature [10].
- A parameterization derived from motion-capture called the Gaze Warping Transformation (GWT) that generates emotionally expressive head and torso movement during gaze shifts [9].

In the EGM, emotionally expressive behavior is performed using the GWT, while the procedural eye model ensures realistic eye behavior in conjunction with the head. The eye movement is automatically generated and layered on top of the head and torso movement. The interaction between the eye and head movement can generate multiple types of gaze, shown in Table 2.

The Gaze Warping Transformation represents the emotional style of a gaze shift. It is determined from the difference between an emotionally expressive gaze shift and an emotionally neutral gaze shift directed towards the same target. It provides two primary benefits: layering and composability. The GWT representation of an expressive behavior can be layered upon an emotionally neutral gaze shift directed toward an arbitrary target that is different from the original gaze shift that generated the GWT. In addition, multiple GWTs that affect different sets of joints can be composed into a single GWT that captures all of the expressive behaviors of the individual GWTs.

In order to drive the SBGC using the EGM, we implemented a set of eXtensible Stylesheet Language Transformation (XSLT) rules compatible with the Nonverbal Behavior Generator (NVBG) [13], another component of the ICT's Virtual Humans Toolkit. XSLT converts one XML-based language into another XML-based language. We use a set of XSLT rules to convert abstract inputs into the specific BML commands used to control the SBGC, using input parameters extracted from the Gaze Warping Transformations.

There are several benefits to connecting the EGM to the SBGC. By connecting the EGM to SmartBody, the EGM, which is an offline animation system, is used to drive a virtual human in real time. SmartBody can then automatically integrate the EGM gaze styles with other controllers, such as those implementing other aspects of nonverbal behavior. In addition, this implementation demonstrates the versatility of the SBGC in producing gaze shifts of various styles, and provides a demonstration of where to obtain the gaze style input parameters for the SBGC.

Table 2: Classes of Gaze Movement Produced by Model

Gaze Type
Eye-Only Shift
Eye-Head Shift
Eye-Head-Torso Shift
Head-Only Movement
Head-Torso Movement

4.1 Procedural Eye Model

Eye movement in the EGM is produced procedurally [10]. This model of eye movement is based on the visual neuroscience literature describing the different movements eyes perform during gaze, and the way in which eye movement and head movement are integrated during gaze shifts [14].

4.1.1 Eye Movement Components

The procedural model of eye movement generates several classes of gaze shifts, shown in Table 2. These gaze classes are generated through the use of the following components:

- Saccades. The saccade is a very rapid, highly-stereotyped eye movement which rotates the eye from its initial position directly to the target;
- Vestibulo-Ocular Reflex (VOR). Through the VOR, the eyes rotate within their orbit so that the gaze maintains the same target while the head moves. It produces the Head-Only and Head-Body movements; and
- Combined Eye-Head Movement. This is used to integrate eye movement and head/torso movement, and generates the Eye-Head and Eye-Head-Body gaze shifts;

The SmartBody Gaze Controller also implements all of these components, although implementations differ between the SBGC and the EGM. By using the implementations available in the SBGC, all of the classes of gaze shift in Table 2 can be replicated using the SBGC.

4.1.2 Generation of Gaze Shift Classes in SBGC

In order to produce gaze shifts representative of each gaze category in Table 2, the joint selection and joint bias parameters of the SBGC are used. In order to produce an Eye-Head-Torso gaze shift, the SBGC can be provided with a target. It will then self-animate a gaze shift using the eyes, the head, and the torso to achieve the target using the default parameters. The Eye-Only and Eye-Head gaze shifts can be generated by using the joint selection of the SBGC to deliberately perform the gaze shift with just the eyes, or with the eyes and the head.

Generating a Head-Only or Head-Torso movement is somewhat different, as these movements are not gaze shifts. Instead, the Head-Only and Head-Torso movements are generated by using the joint bias parameters of the SBGC. By providing a joint bias, the head and torso can rotated relative to the eye target, and the SBGC will automatically keep the eyes locked on target, maintaining the VOR. In addition, by using a joint bias in conjunction with the joint selection parameter, the head can rotate with regard to the gaze target while the eyes and torso maintain their orientation.

4.2 Gaze Warping Transformation

Having reproduced the EGM classes of gaze movement within the SBGC, we can then extract specific values from a GWT to transform those gaze shifts.

A GWT represents the difference between two gaze shifts. By taking motion capture of any two gaze shifts a GWT can be derived from the difference between them, such that if that GWT is applied to the first gaze shift it is transformed into the second gaze shift. A GWT can represent emotional manner: the way in which an emotionally expressive gaze shift differs from an emotionally neutral gaze shift towards the same target. This is done by obtaining two motion captures of gaze shifts directed from the same start point to the same target, one emotionally expressive, the other emotionally neutral, and finding a GWT between them.

That GWT can then be applied to a new emotionally neutral gaze shift towards a new target, and it will layer the emotional style it represents onto that new emotionally neutral gaze shift.

4.2.1 GWT Representation

The GWT is a sparse representation of the difference between two motion-captured gaze shifts. The motion capture data of these shifts are represented as a set of 2D motion curves, where each curve represents the value of a single degree of freedom, such as a joint angle, in the animated body over the number of frames captured in the movement. The motion curves can be represented more sparsely through the use of *keyframes*. The keyframes of an animated movement – represented as a set of (frame, value) pairs – are a subset of the frames for that animation, such that the values of the animation curves for the intermediate frames can be found by interpolating between keyframe values. The GWT transforms the keyframes of one gaze shift into the keyframes of another gaze shift.

A GWT is represented as an $m * n$ set of (c_i, b_i) pairs, where m is the number of animation curves in the animated body and n is the number of key frames in each animation curve. Each pair then represents the difference between the ith keyframes of the two animations that GWT is derived from.

The first element of each pair, c_i is a time scaling factor which represents the temporal difference between the ith keyframes of the two curves. Given two key frames from different movements respectively defined as the (frame, value) pairs (ut_i, u_i) and (vt_i, v_i), the time scaling factor c_i is calculated as the ratio of the number of frames between two keyframes in the first shift to the number of frames between two keyframes in the second shift. For example, if the number of frames between two keyframes in the emotionally neutral gaze shift is 10, and the number of frames between two keyframes in the emotionally expressive shift is 30, then c_i for those keyframes is 3. This allows the velocity to differ between each pair of keyframes, allowing for subtle movement patterns. The formula for c_i is:

$$c_i = \frac{ut_i - ut_{i-1}}{vt_i - vt_{i-1}} \quad (1)$$

The second element, b_i, is a spatial offset parameter that represents the difference between the spatial values of ith key frames of the two curves. It is calculated using:

$$b_i = u_i - v_i \quad (2)$$

The GWT is a point-wise transformation of the key frames. In order to calculate the entire GWT, these operations are repeated for every keyframe of every motion curve of the gaze shift, resulting in the full transformation. The GWT representation used in [9] provided eighteen motion curves. The head, shoulders, and torso of the character each had six motion curves associated with them. Three curves represented the (x, y, z) position of the joint, and three curves provided the Euler orientation of the joint.

4.2.2 Layering and Composability of GWTs

The Gaze Warping Transformation has two properties, layering and composability, that enable it to produce a broad variety of gaze shifts with a minimum of motion capture.

The layering property describes how the GWT can be used to transform gaze shifts to arbitrary targets. Once a GWT has been derived from the difference between the motion captures of two gaze shifts, it can then be used to transform a new, emotionally neutral gaze shift directed at an arbitrary target. This will then layer the desired emotionally expressive behavior on top of the gaze shift while still looking at the new target.

Gaze Warping Transformations can also be composed. If the GWTs are composed through operations such as blending or interpolation, it is not certain that the resulting animation will have the same expressive content of the individual GWTs. However, by limiting the multiple GWTs to those that transform different joints in the skeleton, the resulting animation can portray the same expressive content [11].

Using these processes, a minimal amount of motion capture data can be used to generate a large number of gaze shifts. As an example, in [11], a space of 150 animations was generated by composing the GWTs of the behaviors in Table 3 in all permutations, and then layering those composed GWTs onto all of the gaze movements in Table 2.

Table 3: List of Gaze Warping Transformations

Physical Component	Possible Behaviors
Head Posture	Raised, Neutral, Bowed
Torso Posture	Neutral, Bowed
Movement Velocity	Fast, Neutral, Slow

4.2.3 Extraction of Parameters from GWT

In order to drive the SmartBody Gaze Controller, the values of c_i and b_i in the GWTs are automatically extracted and mapped to the bias and velocity parameters of the SBGC.

In order to drive the joint bias parameters using the GWT, we distill the motion curves that represent each joint angle of the GWT down to a single value using the following formula:

$$p = \max_{0 \le i \le n}(b_i) \quad (3)$$

This takes the maximum offset for each joint angle in the GWT and uses it as the input to the SBGC bias setting. The GWT data representing head, shoulder, and torso joints were used to provide SBGC inputs for the head, chest, and back keys.

The speed parameter of the SBGC also requires the time scaling factors of the GWT to be distilled to a single value. To distill a single velocity parameter v from the GWT, we use the inverse of the average of the time scaling parameters for a motion curve. The velocity of SBGC gaze shifts can then be scaled by this factor.

$$v = \left(\frac{1}{n} \sum_{i=0}^{n} c_i \right)^{-1} \quad (4)$$

4.2.4 Using GWT Parameters to drive SBGC

These parameters could be used to directly control the SBGC by being passed to SmartBody as Behavior Markup Language (BML) commands. However, we have implemented the GWTs as XSLT rules compatible with the NVBG [13]. The XSLT rules automatically produce BML to drive the gaze shifts within the SBGC, allowing for control that is abstracted away from the low level animation parameters. For example, when provided a gaze command using the abstract physical description of a GWT:

```
<gaze target='FORWARD' style=
'GWT_EyeHeadTorso_velocity_fast_head_bowed'
/>
```

The XSLT rules will use the low-level physical parameters extracted from the GWT to produce an actual BML command:

```
<gaze target="FORWARD"
sbm:joint-range="BACK EYES"
sbm:joint-speed="180 180 1000">
<sbm:head pitch="45" pitch-min="45"/>
</gaze>
```

The rules also provide the additional advantage of allowing increased abstraction in the future. While the current set of rules simply provide numerical values for physical behavior, as future research continues to define how the physical display of gaze behaviors is linked with the perception of emotion, XSLT rules based upon this research can be developed.

Figure 4: Example Expressive Gaze

5. RESULTS

We have reproduced all 150 of the animations used in the study reported in [11] using the XSLT rules to drive the SBGC. These animations can be found online.[1] While we have not yet performed a formal evaluation, we have found several interesting results.

First is that the layering and composability properties of the GWT were maintained in the SBGC implementation. The gaze movements in Table 2 are implemented using the horizontal joint bias parameters and the joint selection parameter, while the GWTs in Table 3 are primarily implemented using the vertical joint bias parameters and the velocity parameter. Layering is performed by providing the SBGC with horizontal joint biases of the gaze movement, and vertical biases and velocity from the GWT.

However, this does not work when performing Eye-Only, Head-Only, or Eye-Head gaze shifts. This is because using the joint selection parameter to force the gaze controller to not utilize the head or torso is incompatible with using the gaze controller to display head or torso bias. Instead, by initializing two gaze controllers, each using the joint selection parameter to operate on mutually exclusive joints, one controller can perform the gaze, while the other controller performs the expressive behavior.

Second, the SBGC was capable of producing gaze animations in real time, while the GWT was not. When running both on the same computer, a 2.4 Ghz laptop with 512 Mb of RAM, the SBGC performed gaze animations in real-time, while the Matlab-based optimization IK of the GWT was only capable of producing approximately 3 frames of animation per second.

Finally, the expressive gaze animations revealed that the eyelids are a unique problem. Currently, the eyelids are controlled by the SmartBody facial animation controller, not the gaze controller. The facial animation controller does not move the upper eyelid as the eye rotates. For example, when a person looks up, their upper eyelid pulls back slightly. While normally not important, this did appear to affect the emotional content of a small number of the generated gaze shifts.

6. CONCLUSION

In this paper, we have described the SmartBody Gaze Controller, a real time self-animating system for producing gaze shifts that display a wide variety of styles and behaviors. The flexibility of this gaze controller is provided through a rich set of input parameters that control aspects of movement such as velocity,

[1] http://people.ict.edu/~blance/galleries/AAMAS.html

postural bias, and the selection of joints committed to a particular gaze task. We have also provided a preliminary implementation of a mapping from the Expressive Gaze Model to the SBGC input parameters, demonstrating how to control the input parameters.

There is more work to be done in this area, as not all of the capabilities of the EGM are used by the SBGC system. While all of the EGM-described behaviors can be generated using the SBGC, the inputs for the SBGC require the distillation of the GWT into a small number of single values. Thus, a great deal of movement information within each behavior is discarded. However, driving the animated character directly using the EGM output limits interactivity, due to the need for motion retargeting.

In addition, the EGM does not fully utilize the capabilities of the SBGC. For example, there is no modulation of joint speed in real time, nor is there any manipulation of joint limits. Finally, SmartBody can utilize some periodic information, such as head nods, which currently cannot be represented with the GWT.

While connecting the Expressive Gaze Model to the SmartBody Gaze Controller provides benefits for both the EGM and the SBGC, there is further work to be done on fully utilizing the capabilities of both models in conjunction with each other. In addition, we will be evaluating animations generated through both methods against each other, to ensure that similar emotional expression is attributed to both sets of animations.

7. ACKNOWLEDGMENTS

This work was sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM), and the content does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

8. REFERENCES

[1] Amaya, K., Bruderlin, A., Calvert, T.: Emotion from Motion. *Proceedings of Graphics Interface*. 1996.

[2] Argyle, M., Cook, M.: *Gaze and Mutual Gaze*. Cambridge University Press. 1976.

[3] Exline, R.: Visual Interaction: The Glances of Power and Preference. In Weitz, S. (ed), *Nonverbal Communication: Readings with Commentary*. Oxford University Press, 1974.

[4] Fukayama, A., Ohno, T., Mukawa, N., Sawaki, M., Hagita, N.: Messages Embedded in Gaze of Interface Agents: Impression Management with Agent's Gaze. *Proceedings of SIGCHI, 2002.* 2002.

[5] Gu, E., Badler, N.: Visual Attention and Eye Gaze During Multiparty Conversations with Distractions. *IVA 2006 LNAI* 4133 (2006): 193.

[6] Hill, R., Belanich, Core, M., Lane, Dixon, Forbell, E., Kim, J., Hart: Pedagogically structured game-based training: Development of the Elect-Bilat simulation. In: *Poster at the 25th Army Science Conference*, Orlando, FL. Nov. 2006.

[7] Kazerounian, K.: On the Numerical Inverse Kinematics of Robotic Manipulators. *ASME Journal of Mechanisms, Transmissions and Automation in Design*, vol. 109, no. 3, 1987.

[8] Kendon, A.: Some Functions of Gaze Direction in Two-Person Conversation.: In Kendon, A*., Conducting Interaction: Patterns of Behavior in Focused Encounters*. Cambridge University Press. 1990.

[9] Lance, B., Marsella, S.: "Emotionally Expressive Head and Body Movement During Gaze Shifts," *IVA 2007, LNAI 4722*. 2007.

[10] Lance, B., Marsella, S.: "A model of gaze for the purpose of emotional expression in virtual embodied agents," in *AAMAS 2008*. 199-206. 2008.

[11] Lance, B., Marsella, S.: The Relation between Gaze Behavior and the Attribution of Emotion: An Empirical Study. H. Prendinger, J. Lester, and M. Ishizuka (Eds.): *IVA 2008, LNAI 5208*, pp. 1–14, 2008.

[12] Lee J., Marsella S., Traum, D., Gratch, J., Lance, B.: The Rickel Gaze Model: A Window on the Mind of a Virtual Human. In C. Pelachaud et al. (Eds.): *Proceedings of Intelligent Virtual Agents*, LNAI 4722, September 2007.

[13] Lee J., Marsella S.: Nonverbal Behavior Generator for Embodied Conversational Agents. *Proceedings of Intelligent Virtual Agents*. 2006.

[14] Leigh, R., Zee, D.: *The Neurology of Eye Movements*. New York: Oxford University Press. 2006

[15] Luenberger, D.: *Linear and Nonlinear Programming*. Addison-Wesley, Reading MA,1984.

[16] Monheit, G., Badler, N.: A Kinematic Model of the Human Spine and Torso. *IEEE Computer Graphics and Applications*. v. 11 n. 2. 1991.

[17] Neff, M., Fiume, E.: AER: Aesthetic Exploration and Refinement for Expressive Character Animation. *Proceedings of the 2005 Symposium on Computer Animation*. 2005.

[18] Peters, C., Pelachaud, C., Bevacqua, E., Mancini, M.: A Model of Attention and Interest Using Gaze Behavior. *Intelligent Virtual Agents 2005*. 2005.

[19] Rickel, J., Johnson, W. L.: Animated Agents for Procedural Training in Virtual Reality: Perception, Cognition, and Motor Control. *Applied Artificial Intelligence*. May, 1999.

[20] Shin, H., Lee, J., Shin, S., Gleicher, M.: Computer Puppetry: an Importance-Based Approach. *ACM Transactions on Graphics.* v. 20 n. 2. 2001.

[21] SmartBody >> BML Gaze Element. *SmartBody Wiki.* SourceForge. http://smartbody.wiki.sourceforge.net/BML+Gaze+Element

[22] Swartout, W., Gratch, J., Hill, R., Hovy, E., Marsella, S., Rickel, J., Traum, D.: Toward virtual humans. *AI Magazine* 27(1). 2006

[23] Thiebaux, M., Marshall, A., Marsella, S., Kallman, M.: Smartbody: Behavior Realization for Embodied Conversational Agents. In *AAMAS '08.* 2008.

[24] Vilhjalmsson, H., Cantelmo, N., Cassell, J., Chafai, N., Kipp, M., Kopp, S., Mancini, M., Marsella, S., Marshall, A., Pelachaud, C.: The Behavior Markup Language: Recent Developments and Challenges. *Proceedings of Intelligent Virtual Agents*, LNAI 4722. 2007.