Authoring Directed Gaze for Full-Body Motion Capture

Tomislav Pejsa Daniel Rakita Bilge Mutlu Michael Gleicher

University of Wisconsin-Madison

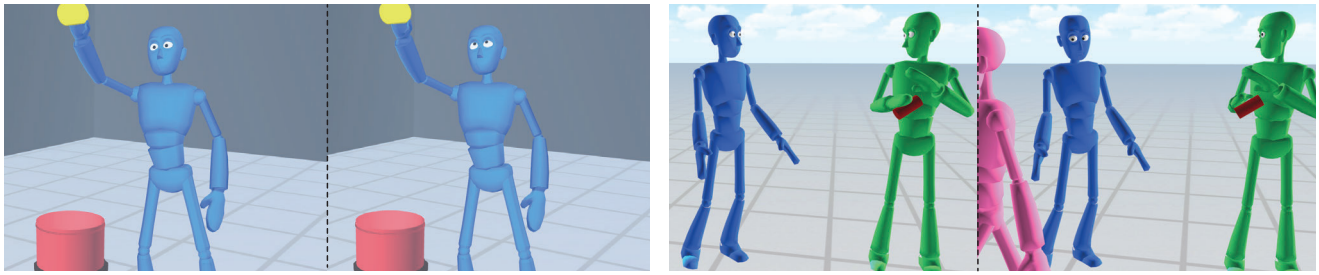

Figure 1: *From left to right: (1) original, captured body motion; (2) original motion with gaze behavior automatically inferred and added by our system; (3) motion with automatically inferred, unedited gaze; and (4) edited motion introducing gaze toward a new character in the scene.*

Abstract

We present an approach for adding directed gaze movements to characters animated using full-body motion capture. Our approach provides a comprehensive authoring solution that automatically infers plausible directed gaze from the captured body motion, provides convenient controls for manual editing, and adds synthetic gaze movements onto the original motion. The foundation of the approach is an abstract representation of gaze behavior as a sequence of gaze shifts and fixations toward targets in the scene. We present methods for automatic inference of this representation by analyzing the head and torso kinematics and scene features. We introduce tools for convenient editing of the gaze sequence and target layout that allow an animator to adjust the gaze behavior without worrying about the details of pose and timing. A synthesis component translates the gaze sequence into coordinated movements of the eyes, head, and torso, and blends these with the original body motion. We evaluate the effectiveness of our inference methods, the efficiency of the authoring process, and the quality of the resulting animation.

Keywords: animation, gaze, synthesis, editing, motion capture

Concepts: •**Computing methodologies** → **Animation;**

1 Introduction

Directed gaze is the intentional movement of a character's line of sight toward targets in space. It is an important component of good character animation, as it can situate characters in the scene, signal the focus of their attention, and convey their personality and intent. Directed gaze involves coordinated movements of the eyes, head, and torso. In animation practice, these movements are usually hand-authored, which requires considerable effort and skill due to their intricate spatial and timing relationships. Performance capture is seldom a viable option; even if the capture rig can record eye movements, the recorded gaze typically does not match the virtual scene and needs to be edited by hand.

Our goal is to facilitate the process of creating character animation with gaze by automatically adding *editable* directed gaze to a captured motion, and by providing convenient controls for its editing. We introduce an abstract model of gaze motion as a sequence of *gaze instances*, where each instance specifies a gaze shift toward a target in the scene. This model serves as the foundation of the authoring process. The workflow in our approach begins with an automatic *inference* of the gaze instance sequence from the body motion and scene layout. The body motion implicitly contains the head and torso components of the actor's gaze, and we can infer a plausible reconstruction of the original gaze behavior by analyzing the motion's kinematic properties. From this sequence, our approach can synthesize the eye movements that match the body motion and scene.

The inferred sequence also serves as a starting point for the *editing* process. The gaze instance model has two key properties that enable convenient editing. First, it abstracts the details of gaze shift timing and pose. An animator can specify *when* and *where* to look by adding, removing, and modifying gaze instances, while the kinematics of eye, head, and body movements are synthesized automatically. Second, the model anchors the gaze motion in the virtual scene for easy spatial editing. Gaze targets serve as editing handles attached to objects and characters. When animators manipulate these handles, synthesized gaze motion automatically adapts to the new spatial constraints. We provide a graphical tool which exposes the operations for editing gaze instances and targets. The inferred and (potentially) edited gaze motion is *synthesized* automatically by a procedural, neurophysiology-based gaze controller, the output of which is blended with the original body motion.

The primary contribution of this work is the gaze *inference* approach (Section 3), which infers an editable model of the character's gaze behavior from motion capture data and scene properties. Other contributions include an approach for *editing* the inferred gaze behavior by modifying the gaze instances and scene layout (Section 5) and methods for *synthesis* of the final motion with gaze (Section 4).

SA '16 Technical Papers,, December 05-08, 2016, Macao
ISBN: 978-1-4503-4514-9/16/12$15.00
DOI: http://dx.doi.org/10.1145/2980179.2982444

ACM Reference Format
Pejsa, T., Rakita, D., Mutlu, B., Gleicher, M. 2016. Authoring Directed Gaze for Full-Body Motion Capture. ACM Trans. Graph. 35, 6, Article 161 (November 2016), 11 pages. DOI = 10.1145/2980179.2982444 http://doi.acm.org/10.1145/2980179.2982444.

2 Related Work

Gaze Synthesis – Researchers have proposed a number of synthesis models for gaze shifts (see Ruhland et al. [2015] for a survey of prior work on gaze in computer animation.). While some models (e.g., Deng et al. [2005]; Lee et al. [2002]) focus on saccades (rapid eyeball movements in gaze), others synthesize coordinated movements of the eyes, head, and body toward targets. For example, Lance et al. [2010] and Heck [2007] use data-driven approaches, while Peters et al. [2010] and Pejsa et al. [2015] introduce procedural models inspired by neurophysiological observations. The procedural models by Thiebaux et al. [2009] and Pejsa et al. [2015] afford parametric control over both the head and torso posture. We chose to extend the latter model, as it is designed to provide biologically plausible movements across the parametric range.

Gaze Inference – We use the term "gaze inference" to describe methods that analytically determine when and where the character should look and generate the corresponding gaze behavior. Research on gaze control in cognitive science, surveyed by Henderson [2003], has shown that people's gaze is influenced by two mechanisms: the spontaneous *bottom-up* attention, which directs their gaze toward visually salient environment features, and the deliberate *top-down* attention, which directs it toward task-relevant objects. Much of prior work has focused on simulating bottom-up attention, typically to synthesize believable idle gaze for virtual agents and avatars. In these approaches, gaze locations are often determined using a saliency map of the scene and computed from features such as color, contrast, orientation, and motion [Peters and O'Sullivan 2003; Peters and Itti 2008]. We perform gaze target inference using a similar representation. In other work, gaze targets are determined from spatial and kinematic properties of scene entities, such as proximity, velocity, and orientation [Cafaro et al. 2009; Grillon and Thalmann 2009; Kokkinara et al. 2011].

In contrast with bottom-up approaches, models of top-down attention generate gaze behaviors based on the character's goals, intentions, and knowledge. Examples include the Rickel Gaze Model [Lee et al. 2007], models of gaze for virtual demonstrators [Huang and Kallmann 2016], and the gaze model for object manipulation tasks by Bai et al. [2012]. Prior work has also introduced hybrid approaches that integrate bottom-up gaze with top-down, task-driven patterns [Khullar and Badler 2001; Mitake et al. 2007]. Our gaze inference approach differs from the preceding work in that its main objective is to support motion editing. As such, it discovers an editable specification of the gaze behavior already encoded in the body motion, rather than generating novel gaze behaviors that potentially overwrite existing gaze movements. The approach is more closely related to top-down models, as it attempts to infer the actor's intent expressed in gaze and encoded in the body motion.

Motion Synthesis and Editing – Our approach can best be characterized as model-based editing of gaze motion. Such approaches use an abstract model to specify what the motion should be and produce the final motion that expresses the author's intent using methods such as spacetime optimization, physics-based modeling, and synthesis from priors. Examples of model-based editing include physics-based spacetime methods [Popović and Witkin 1999], motion path editing [Gleicher 2001], spatial-relationship-preserving interaction editing [Ho et al. 2010], data-driven hand animation for guitar playing [ElKoura and Singh 2003], among others.

Our work is also related to automatic methods for adding missing movements to motion capture data. Several methods have been introduced for inferring hand movements from a given body motion [Jörg et al. 2012; Ye and Liu 2012], but our work is the first to consider the analogous problem of inferring eye gaze.

3 Gaze Inference

The gaze inference component of our approach analyzes the input body motion and virtual scene and infers a sequence of gaze instances that serves as a model of gaze motion. Each gaze instance is a tuple $G = (f_s, f_x, f_e, T, \alpha_H, \alpha_T)$, where f_s, f_x, and f_e are gaze shift start frame, fixation start frame, and fixation end frame, respectively, and T is the gaze target. α_H and α_T are *alignment* parameters, which specify the head and torso posture relative to the target (Figure 4)—these parameters serve as inputs into the gaze controller (Section 4) and enable convenient control over posture during editing.

Gaze inference begins with signal analysis of the body motion to detect individual gaze instances and infer their timings, f_s, f_x, and f_e. The inferred sequence may contain gaps, during which no directed gaze is applied to the character—we refer to such gaps as *unconstrained gaze*. Next, we analyze the character's view of the scene at the end of each gaze shift and infer the most probable target T. Finally, we compute the alignment parameter values, α_H and α_T, from the head and torso postures at the end of each gaze shift.

3.1 Gaze Instance Inference

Human gaze shifts follow a particular kinematic pattern. The eyes, head, and torso accelerate toward the gaze target, reach their peak velocities at the midpoint of the gaze shift, and decelerate as they align with the target, at which point the fixation begins [Uemura et al. 1980; Fuller 1992; McCluskey and Cullen 2007]. We detect gaze shifts by searching for this pattern in the body motion using a three-step approach. First, we search for clusters of maxima in angular acceleration signals of the head and torso joints, which correspond to significant gaze events (gaze shift beginnings and ends). Coleman et al. [2008] used a similar method for key pose extraction. Each pair of adjacent gaze events defines the boundaries of a motion interval, which can be either a gaze shift or a fixation. In the second step, we classify the motion intervals into gaze shifts and fixations. Third, for each adjacent gaze shift-fixation pair, we generate a gaze instance.

Gaze Event Discovery – We discover gaze events by analyzing joint accelerations in the motion (Figure 2). Let $a_j(f)$ be the angular acceleration signal of the joint j, defined across the motion's frame range ($f \in [0, L\rangle$, where L is motion length in frames). We normalize the acceleration signals to the 0-1 range: $\hat{a}_j(f) = a_j(f) / max(a_j(f))$. We use the normalized acceleration signals of the root, torso, and head joints as local evidence measures indicating probability of significant gaze events (i.e., gaze shift beginning or ending). We compute the global probability of a gaze event as a weighted sum of normalized acceleration signals:

$$p_E(f) = \frac{\sum_{j \in J} w_j \hat{a}_j(f)}{\sum_j w_j} \quad (1)$$

where J is the set of joints, consisting of the root, torso, and head joints. Weights w_j define each joint's influence on the global probability. We heuristically define a joint's weight to be greater the closer it is to the eyes, i.e., the head contributes more to the global probability than the torso or root. This choice is based on findings showing the importance of head movements for the execution and perception of human gaze [Hietanen 1999]. We calculate joint weights as $w_j(f) = 1/(1 + l_j)$, where l_j is the distance between joint j and the eyes, measured as the cumulative length of the segments between them.

Local maxima in p_E correspond to the beginnings and endings of gaze shifts. Since acceleration signals are likely to be noisy and contain a large number of spurious maxima, a bilateral filter is applied to p_E to smooth out the spurious maxima while preserving

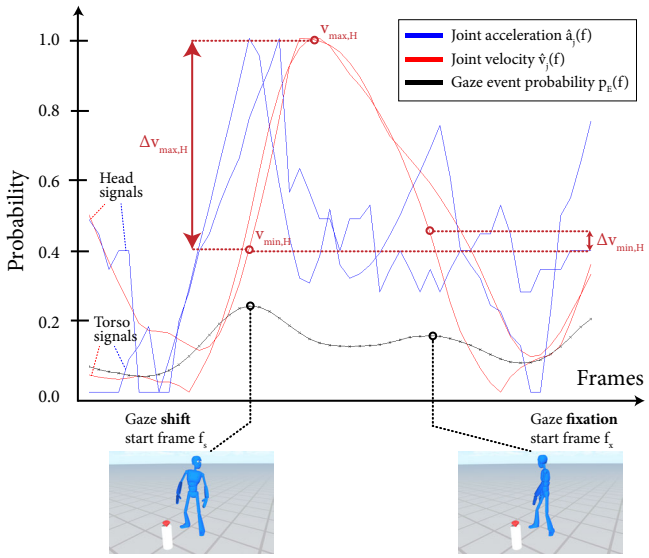

Figure 2: *Gaze shift detection through analysis of acceleration and velocity signals of the head and torso. We first aggregate the normalized joint accelerations $\hat{a}_j(f)$ into a global probability signal $p_E(f)$. Adjacent local maxima in $p_E(f)$ define the boundaries of a motion interval. We determine that the interval is a gaze shift from its high peak velocity, $v_{\max,j}$. Velocity signals in the graph were normalized for easier viewing, whereas inference is performed on unnormalized velocities.*

those that are kinematically significant. We set the filter's range parameter σ_r to 0.5 and spatial parameter σ_s to 3. The filtered probability signal is then searched for local maxima. Each pair of adjacent maxima defines a motion interval $I_k = (f_{s,k}, f_{e,k})$, which could either be a gaze shift or a fixation.

Motion Interval Classification – To determine if a motion interval I_k is a gaze shift, we use the fact that gaze shifts are characterized by peaks in joint velocity (Figure 2) and we construct another probabilistic measure. Let $v_j(f)$ be the angular velocity signal of joint j. We define $v^k_{s,j}$ and $v^k_{e,j}$ to be joint velocities at the boundaries of the interval I_k. Furthermore, let $v^k_{\max,j}$ and $v^k_{\min,j}$ be the maximum and minimum joint velocity over the interval I_k. Then we define maximum and minimum velocity differences as $\Delta v^k_{\max,j} = v^k_{\max,j} - min(v^k_{s,j}, v^k_{e,j})$ and $\Delta v^k_{\min,j} = max(v^k_{s,j}, v^k_{e,j}) - v^k_{\min,j}$. Finally, we define the ratio of these quantities as follows:

$$r^k_j = \begin{cases} 0 & \text{if} \Delta v^k_{\max,j} \leq 0 \\ \frac{\Delta v^k_{\max,j}}{\Delta v^k_{\min,j}} & \text{otherwise} \end{cases} \quad (2)$$

The ratio r^k_j increases with peak velocity relative to the velocities at interval boundaries, indicating a gaze shift at the interval. A logistic function then maps the ratio to a probability value in the 0–1 range:

$$p^k_{S,j} = \frac{2}{1 + exp(-r^k_j)} - 1 \quad (3)$$

Finally, we compute the gaze shift probability as a weighted sum of per-joint probabilities:

$$p^k_S = \frac{\sum_{j \in J} w_j p^k_{S,j}}{\sum_j w_j} \quad (4)$$

where w_j are the joint weights defined previously.

We classify the interval I_k as a gaze shift if $p^k_S > P_{S,\min}$ (where $P_{S,\min}$ is a probability threshold set to 0.6) and as a fixation otherwise. Adjacent intervals that are classified as fixations get aggregated into a single fixation. We end up with a sequence of gaze shift-fixation pairs, from which we generate the sequence of gaze instances.

Gaze Instance Generation – For each gaze shift-fixation pair (I_k, I_l), we generate a gaze instance $G = (f_s, f_x, f_e)$, such that the gaze shift starts at $f_s = f_{s,k}$, while the fixation starts at $f_x = f_{s,l}$ and ends at $f_e = min(f_{e,l}, f_{s,l} + L_{\max} - 1)$. Note that the fixation may not be longer than $L_{\max}$ (set to 1 second by default)–this is to prevent gaze from becoming "stuck" on a target for long periods of time.

3.2 Gaze Target Inference

In the preceding step, we inferred all the gaze instances and their timings. Next, for each gaze instance we infer its target in the scene. We assume there exists at least a simplified 3D model of the scene containing the key characters and objects, like those used for previsualization. The result of gaze target inference is a sequence of gaze instances with associated targets. Gaze targets are handles in the scene attached to objects and other characters, which the animator can move around to change where the character is looking.

Our gaze inference method is based on three heuristics. First, the character is more likely to be looking at a point that lies along the movement direction of its head. If the character turns right, it is unlikely that the target lies to the left, up, or down. Second, the character is more likely to look at objects and characters that are *important* to the story or task. For instance, in a scene where the character steals a gem, the gem is a likely gaze target. We let the animator manually label scene objects as important. Finally, the character is more likely to gaze at objects just before touching or picking them up [Johansson et al. 2001]. Assuming hand contacts are annotated in the body motion, we can easily determine when the character is about to handle an object.

We use a probabilistic approach for gaze target inference. Recall the gaze instance specification from the preceding step: $G = (f_s, f_x, f_e)$. The gaze fixation start time, f_x, is when the character's eyes are aligned with the target; therefore, the target must be within the character's view. We compute a probability distribution, $p_T(\mathbf{x})$, over the 2D projection space representing the character's view of the scene at the frame f_x. Here $\mathbf{x}$ denotes a point in the 2D projection space. For every point in the character's view, $p_T(\mathbf{x})$ gives us the probability that the character is gazing at it. We find the gaze target T at the global maximum of p_T.

We model $p_T(\mathbf{x})$ as a weighted sum of three probability terms, each implementing one target inference heuristic:

$$p_T = \frac{w_d p_d + w_i p_i + w_h p_h}{w_d + w_i + w_h} \quad (5)$$

$p_d(\mathbf{x})$ is the directional term, which assigns higher probabilities to points lying along the direction of head movement. $p_i(\mathbf{x})$ is the importance term and is non-zero at points corresponding to objects and characters labeled as important. p_h is the hand contact term, which is non-zero for objects that are about to be handled by the character. These terms contribute to the final probability, p_T, in proportion to their weights, w_d, w_i, and w_h. We empirically find that the best inference results are obtained by setting these weights to 0.6, 0.15, and 0.25, respectively. Given the final probability p_T, we obtain the gaze target T at the global maximum of p_T: $\mathbf{x}_T = argmax_{\mathbf{x}}\, p_T(\mathbf{x})$.

Figure 3 shows an example scene view with the associated probability distributions and gaze target location indicated. In the remainder

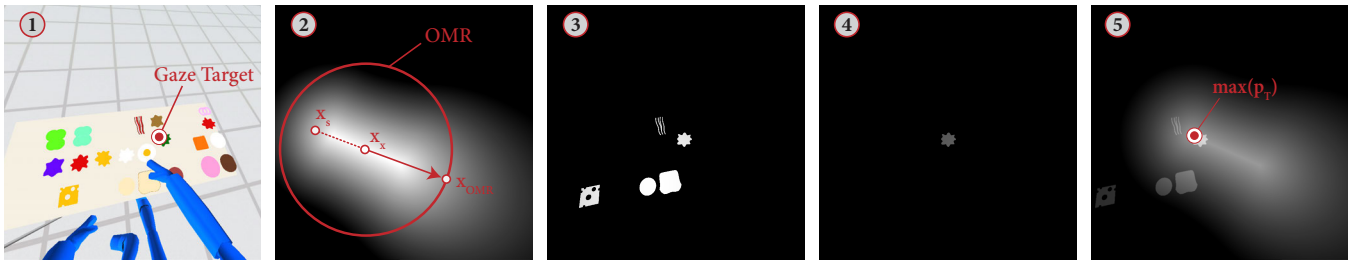

Figure 3: *We infer the gaze target by analyzing the character's scene view at the end of the gaze shift (1). The gaze target is more likely to be located along the head movement direction (directional term, 2), on an object labeled as important (importance term, 3), or on an object that is about to be touched or picked up (hand-contact term, 4). We average these probability terms and find the target at the global maximum (5).*

of this section, we provide more detail about the computation of individual probability terms.

Directional term – We compute the directional term, $p_d(\mathbf{x})$, as follows. When the character shifts its gaze, its head traces a path in the projection space beginning at $\mathbf{x}_s$ and ending at $\mathbf{x}_x$ (Figure 3-2). If the character always gazed straight on, then $p_d(\mathbf{x})$ would be highest at $\mathbf{x}_x$. However, the character is as likely to be looking at something out of the corner of the eye, so the eyes may "overshoot" $\mathbf{x}_x$ and line up with a point further along the movement path. Therefore, from the path $(\mathbf{x}_s, \mathbf{x}_x)$ we extrapolate a path extending to the point $\mathbf{x}_{\text{OMR}}$, which lies on the edge of the eyes' motor range (OMR), beyond which the eyes cannot move (Figure 3b). $p_d(\mathbf{x})$ has the maximum value of 1 along the path $(\mathbf{x}_x, \mathbf{x}_{\text{OMR}})$ and linearly falls off to zero with distance from the path. Note that we do not set p_d to zero everywhere outside the OMR region. Due to variability in human OMR and differences between the capture environment and the virtual scene, the gaze target may actually lie outside the character's OMR. By allowing non-zero probability beyond that range, we allow our algorithm to pick up the gaze target even in such cases.

Importance term – The importance term, $p_i(\mathbf{x})$, increases the probability of placing gaze target handles on important objects and characters (Figure 3-3). Let Ω_i be the (possibly discontiguous) region in the projection space occupied by projections of all the visible, important objects. The importance term is zero outside that region, while within the region it is computed as: $p_i(\mathbf{x}) = \mid \mathbf{v} \cdot \mathbf{n} \mid$, where $\mathbf{v}$ is the vector from the eye centroid to the current projection point $\mathbf{x}$ and $\mathbf{n}$ is the object's surface normal at that point. We use the dot product to increase the probability of placing gaze target handles on the more salient parts of the object's surface rather than along the silhouette.

Hand contact term – The hand contact term, p_h, increases the probability of placing gaze target handles on an object that the character is about to pick up or touch. As shown in studies of human eye-hand coordination [Johansson et al. 2001], gaze toward the object is most likely to occur 1 second before hand contact, though it may begin 3 seconds before and end up to 0.6 seconds after. We use this rule to compute our probability term. If t_x is the current time in the motion (corresponding to fixation start frame f_x) and t_h is the start time of a hand contact with an object (obtained from hand contact annotations), then probability of gaze toward that object is:

$$p_{h,0} = \begin{cases} 0 & t_x < t_h - 3 \\ \frac{t_x - t_h + 3}{2} & t_x \geq t_h - 3 t_x < t_h - 1 \\ \frac{t_x - t_h + 1}{1.6} & t_x \geq t_h - 1 t_x < t_h + 0.6 \\ 0 & t_x \geq t_h + 0.6 \end{cases} \quad (6)$$

We define Ω_h to be the region of the projection space occupied by the handled object. The hand contact probability term (Figure 3-4) is zero outside that region, while within the region we compute it as $p_h(\mathbf{x}) = p_{h,0} \mid \mathbf{v} \cdot \mathbf{n} \mid$. This is similar to the importance term, except that maximum probability is scaled by $p_{h,0}$.

Implementation – To achieve reasonable computation time, we compute gaze target probabilities on the GPU. We render the scene in multiple passes using a wide-angle camera located at the character's eyes and pointed in the same direction as the head. On each pass, we compute one of the probability terms p_d, p_i, and p_h in a shader program. We blend the terms to get the final probability p_T. We then retrieve the render texture containing p_T and do a linear search for $\mathbf{x}_T$. Finally, we use a raycast through $\mathbf{x}_T$ to determine the gaze target's 3D position and the scene object to which it is attached.

3.3 Computing Alignment Parameters

Having inferred all the gaze instances and their targets, we also compute the values of the head and torso alignment parameters, α_H and α_T. When humans shift their gaze, they may also reorient their head or torso toward the target, which affects how the gaze shift is perceived by others. For example, looking at something out of a corner of the eye signals a lower level of attention than when turning the head to face the target fully.

Our gaze synthesis approach (Section 4) allows the animator to control the amount of head and torso reorientation using the alignment parameters, α_H and α_T, and thus obtain a greater variety of gaze shifts. Parameter values range from 0 to 1, where 0 specifies minimal reorientation, while 1 fully aligns the head or torso with the target (Figure 4). During gaze inference, we compute the values of the alignment parameters that match the postures encoded in the original motion. The animator can then edit the postures by hand-editing the alignment values (Section 5).

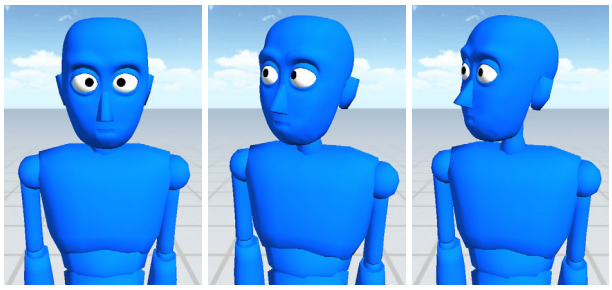

Figure 4: *Effects of the head alignment parameter on posture at the end of the gaze shift. Left: initial head and torso posture before the gaze shift. Middle: gazing toward the target with minimal head alignment ($\alpha_H = 0$). Right: gazing toward the same target with full head alignment ($\alpha_H = 1$).*

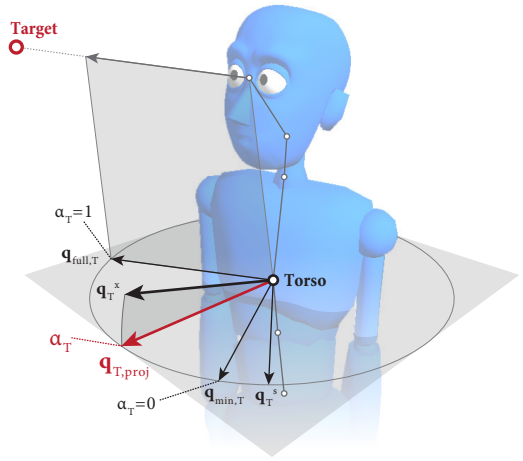

Figure 5: *Computing torso alignment parameter α_T for a gaze shift starting at frame f_s and ending at frame f_x.*

We can analytically estimate the head and torso alignment parameters once the timing and target of a gaze shift are known. Figure 5 illustrates the procedure for computing the alignment parameter α_T, which we compute from orientations of the uppermost torso joint at the start and end of the gaze shift. Let f_s be the start frame and f_x the end frame of the gaze shift. The torso rotates from the orientation $\mathbf{q}_T^s$ to $\mathbf{q}_T^x$. We set the character to the start pose and compute $\mathbf{q}_{\text{full},T}$, which is the orientation which would fully align the torso joint with the target (corresponding to $\alpha_T = 1$), and $\mathbf{q}_{\text{min},T}$, which is the minimal torso orientation (corresponding to $\alpha_T = 0$). $\mathbf{q}_{\text{min},T}$ depends on gaze shift amplitude: it is zero for low-amplitude gaze shifts, but increases monotonically with amplitude when the latter is above 20°. We refer the reader to Pejsa et al. [2015] for more detail about computing $\mathbf{q}_{\text{min},T}$.

Next, we project $\mathbf{q}_T^x$ onto the arc defined by orientations $\mathbf{q}_T^s$ and $\mathbf{q}_{\text{full},T}$—we denote this projected orientation $\mathbf{q}_{T,\text{proj}}$. Finally, we compute α_T as:

$$\alpha_T = \frac{\angle(\mathbf{q}_{T,\text{proj}}, \mathbf{q}_{\text{min},T})}{\angle(\mathbf{q}_{\text{full},T}, \mathbf{q}_{\text{min},T})} \tag{7}$$

The procedure for computing α_H is analogous.

3.4 Evaluation

The purpose of gaze inference is to infer editable gaze behaviors that fit the character's body motion and scene by (1) detecting key gaze shifts in the original motion and (2) inferring gaze target handles on important objects. We evaluated the effectiveness of the gaze inference component on both criteria, while using motion capture, eye tracking, and human annotation data as ground-truth.

Data Collection – We acquired ground-truth body motion and eye tracking data using a Motion Analysis optical motion capture system and SMI Eye Tracking Glasses. We recorded nine scenes, in which the actor engaged in activities with rich gaze behaviors: environment interactions, locomotion, two-person conversation, and so on. The cumulative length of all the motions was 6:46 minutes. Next, we had two human coders annotate *gaze shifts* and *important gaze targets* in each scene. For gaze shift annotations, the coders were instructed to visually analyze body motion kinematics and mark gaze shift start and end times. For gaze target annotations, we selected five scenes that involved extensive gaze toward important objects, with a total length of 4:57 minutes. From the eye tracker video, we extracted representative image frames for each fixation detected by the eye tracker. Finally, we instructed the coders to annotate the gaze target in each image. For example, for the StealGem scene, any frame showing gaze toward the gem object was marked "Gem."

Scene	Sensitivity	$r(t_s)$	$r(t_e)$
WindowWashing	86%	1.00	1.00
WalkCones	58%	0.99	0.99
BookShelf	84%	1.00	1.00
StealDiamond	80%	1.00	0.99
WaitForBus	97%	1.00	1.00
StackBoxes	69%	1.00	1.00
MakeSandwich	100%	1.00	1.00
MakeSandwichDemo	81%	1.00	1.00
ChatWithFriend	58%	1.00	1.00

Table 1: *Results of gaze instance inference evaluation.*

Results: Gaze Instances – Out of the total motion dataset, 2:30 minutes were annotated by both coders, allowing us to compute intercoder agreement. We first identified pairs of overlapping gaze shift annotations in both datasets. We found that 81% of the gaze shifts in the first dataset had a match in the second dataset, and 90% of the gaze shifts in the second dataset had a match in the first dataset. Next, we measured correlation of gaze shift start times and end times between the two datasets using Fisher's formula for intraclass correlation [Fisher 1925]. We obtained $r = 1.00$ for both start and end times, indicating very high correlation between the coders' estimates of gaze shift timings.

We used a similar approach to measure the accuracy of gaze instance inference. First, we identified pairs of overlapping gaze shifts in the inference output and ground-truth annotations. Next, we computed the *sensitivity* of gaze instance inference, which is the percentage of ground-truth gaze shifts that had a match in the inference output. This measure is significant because it tells us how effective the inference method is at picking up key gaze shifts. Finally, for the pairs of overlapping gaze shifts, we computed Fisher's r as a measure of correlation of their start times (t_s) and end times (t_e), respectively. Table 1 shows the evaluation results for all nine scenes. For the ChatWithFriend scene we found that our method achieved relatively low sensitivity. This scene depicts a character standing and conversing with another person. Most of the gaze shifts are conversational gaze aversions involving subtle head movements, which may be difficult to detect using our method.

Results: Gaze Targets – As with gaze shift annotations, we had our coders independently annotate an overlapping portion of the dataset that was comprised of three scenes with a total length of 2:32 minutes. We calculated Cohen's κ as the measure of intercoder agreement; κ ranged from 0.93 to 1.00, indicating very high agreement.

Next, we evaluated the accuracy of gaze target inference as follows. Our analysis included only gaze instances toward important scene objects and characters. For each gaze instance in the inference output, we searched for overlapping fixations in the ground-truth data. If there was at least one fixation where the target label matched the inferred target for that instance, we counted the instance as correct. The percentage of correctly inferred instances served as inference accuracy. Accuracies for the five scenes were as follows: BookShelf (63%), StealDiamond (100%), StackBoxes (50%), ChatWithFriend (83%), and MakeSandwichDemo (82%). Low accuracy of the Stack-

Boxes scene may be due to errors in ground-truth data, as the scene involved downward glances toward boxes carried by the actor, which may have lain beyond the field of view of the eye tracker camera.

4 Gaze Synthesis

Given a gaze behavior specified as a sequence of instances, the gaze synthesis component produces a gaze motion and adds it to the character's body motion. To synthesize plausible gaze shift movements toward each target, we adapt a procedural gaze controller [Pejsa et al. 2015]. The controller is driven by neurophysiology-derived kinematic laws and has been empirically shown to produce natural-looking gaze shifts, while affording significant parametric control over their kinematic properties—target position, head and torso posture, and velocity. In this section, we give an overview of gaze shift kinematics and our modifications required to support highly varied body motions. We also describe a blending scheme for combining the controller output with the original body motion.

4.1 Synthesis of Gaze Kinematics

The gaze controller is a feed-forward system generating rotational motion of the eyes, head, and torso joints toward a gaze target T. A more detailed description of the controller's operation is provided by Pejsa et al. [2015]—here, we highlight the key functionality and our extensions.

Kinematics of eye, head, and torso movements are described by velocity profiles parametrized by their peak velocities, $v_{\max,E}$, $v_{\max,H}$, and $v_{\max,T}$. Peak velocities depend linearly on gaze shift amplitude—the further the joints need to rotate to reach the target, the faster they move. In general, peak velocities are computed using equations of the form $v_{\max,*} = (a_* D_* + b_*)v_{0,*}$, where $*$ denotes the body part (eye, head, or torso), while a_*, b_*, and $v_{0,*}$ are constants specific to the body part (values provided by Pejsa et al. [2015]). D_* is the body part's rotational amplitude, which brings it into alignment with the target. Two other important kinematic properties are head and torso latency times, τ_H and τ_T. The head and torso usually do not begin moving toward the target at the same time as the eyes, but lag behind by their respective latency. Latencies also linearly depend on gaze shift amplitude: $\tau_* = c_* D_* + + d_*$, where c_* and d_* are per-body-part constants.

To synthesize the pose at a frame f, at which there is an active gaze instance $G = (f_s, f_x, f_e, T, \alpha_H, \alpha_T)$ ($f_s \leq f \leq f_e$), we invoke the gaze controller and supply it the target position, $\mathbf{p}_T$, and head and torso alignments, α_H and α_T. $\mathbf{p}_T$ determines the direction in which the body parts will rotate, while the alignment parameters determine *how much* the head and torso contribute to the overall movement (as described in Section 3.3). From these parameters, the controller computes the target posture needed to bring each body part into alignment with the given target. Let $\mathbf{q}_H^s$ and $\mathbf{q}_T^s$ be the orientations of the topmost head and torso joints at the start of the gaze shift. The controller computes the target posture as orientations $\mathbf{q}_H^x$ and $\mathbf{q}_T^x$. From these orientations it computes the rotational amplitudes: $D_* = \angle(\mathbf{q}_*^s, \mathbf{q}_*^x)$, which it uses to calculate peak velocities and latencies. Having computed the required kinematic parameters, the controller synthesizes the gaze shift as rotational movements from $\mathbf{q}_*^s$ to $\mathbf{q}_*^x$, as described by Pejsa et al. [2015].

Root Movement Adaptation – Since key kinematic parameters (velocities and latencies) are computed from amplitude, the gaze shift's kinematics very much depend on the target position relative to the character. When adding a gaze shift to a character that is moving relative to the target (e.g., walking and turning), we must compute these parameters based on the target's projected position at the *end* of the gaze shift to avoid implausible gaze shift kinematics. Our solution is to supply the gaze controller with an additional parameter $\Delta\mathbf{p}_T$—the *relative target translation* due to root movement over the course of the gaze shift. We apply this translational offset to the target position: $\mathbf{p}'_T + \mathbf{p}_T + \Delta\mathbf{p}_T$. The adjusted target position $\mathbf{p}'_T$ is used for computing initial kinematic parameters—amplitudes, peak velocities, and latencies—resulting in more plausible kinematics.

Computing $\Delta\mathbf{p}_T$ is straightforward. Let $\mathbf{p}_{R,0}$ and $\mathbf{q}_{R,0}$ be the world-space position and orientation of the character's root at the start of the gaze shift (frame f_s). Let $\mathbf{p}_{R,1}$ and $\mathbf{q}_{R,1}$ be the root position and orientation at the end of the gaze shift (frame f_x). Finally, let $\mathbf{p}_T$ be the gaze target position. Relative target translation is then: $\Delta\mathbf{p}_T = \mathbf{p}_{R,0} + (\mathbf{q}_{R,1}^{-1}\mathbf{q}_{R,0})(\mathbf{p}_T - \mathbf{p}_{R,1}) - \mathbf{p}_T$.

4.2 Gaze Motion Blending

At each frame f, the gaze controller outputs a head and torso posture as a set of joint orientations, $\mathbf{q}_j^f$, where j is the joint index. We blend this posture with the character's current pose (encoded in the original motion), $\mathbf{q}_{0,j}^f$. Thus we get the blended pose: $\mathbf{q}_{\text{blend},j}^f = slerp(\mathbf{q}_{0,j}^f, \mathbf{q}_j^f, w^f)$. To compute the blend weight, w^f, we devised a scheme that achieves two properties: (1) preserves as much of the original motion as possible, and (2) ensures smooth transitions from unconstrained gaze to constrained gaze and vice versa. Therefore, the blend weight has two components: $w^f = w_1^f w_2^f$.

We compute the first component using the cosine of the angle between the original and gaze orientation at the current frame: $w_1^f = 1 - \cos\phi^f$, where $\phi^f = \angle(\mathbf{q}_{0,j}^f, \mathbf{q}_j^f)$. We clamp this value to 1 for $\phi^f > 90$ deg. If a gaze instance is unedited, then the output gaze direction from the controller will be approximately the same as in the original motion, the blend weight will be close to 0, and the original pose will be unmodified. On the other hand, if we change the gaze target for the instance, gaze controller output will diverge from the original motion and its influence on the final pose will increase, allowing the character's gaze to reach the new target. This scheme allows us to automatically control the trade-off between procedurally synthesized and the more expressive, original motion data. We also allow the animator to fix or keyframe the value of w_1 and thus exercise more control over the result.

The second blend weight component, w_2^f, ensures smooth transition at the boundaries between constrained and unconstrained gaze. Gaze instance specification can contain gaps between two adjacent instances. During these gaps, no gaze is synthesized and the original motion is unchanged. When we have a gaze instance, $G = (f_s, f_x, f_e, \ldots)$ following such a gap, we need to smoothly blend in gaze controller output: $w_2^f = -2t^3 + 3t^2$, where $t = (f - f_s)/(f_e - f_s)$. Similarly, we employ the inverse scheme to blend out of a gaze instance preceding a gap.

The blended pose $\mathbf{q}_{\text{blend},j}^f$ is also the final pose, unless the body part is the torso and there are active end-effector constraints. If torso movements introduced by the controller violate the constraints, we correct the torso pose using an IK solver adapted from prior work [Shin et al. 2001].

5 Gaze Editing

The gaze inference component (Section 3) estimates a model of the gaze behavior, which is sufficient to synthesize a plausible gaze motion for the current body motion and scene. However, the animator may wish to make further edits, such as correcting errors in the actor's performance, changing the layout of scene objects,

and adjusting gaze to express different personality and intent. Such edits are challenging to accomplish using a traditional keyframing workflow. We make them easier by relying on a model of gaze that abstracts away kinematic details of eye, head, and torso movements, and provides a set of convenient gaze editing operations. We also provide a graphical tool for performing these operations.

5.1 Editing Operations

Our approach allows the following editing operations: (1) editing where the character is looking by moving gaze targets or scene objects to which they are attached; (2) directly editing the parameters of specific gaze instances, such as setting a different gaze target; (3) adjusting head and torso posture using alignment parameters; (4) changing the timing (start and end frames); (5) adding new gaze instances; and (6) removing existing ones. For all these operations, the synthesis component (Section 4) automatically synthesizes the gaze animation that matches the new specification.

Implementing operations 1–3 is straightforward, as they change specific properties of the gaze instances, while changes to the animation are effected by the synthesis component. Operations 4–6 change the timing and content of gaze instances, so we designed them to preserve the validity of the gaze sequence. Two gaze instances may not overlap, so when adding or extending a gaze instance, we either trim or remove the adjacent gaze instances. Gaze instances must have a minimal length (0.3 seconds by default), as executing a gaze shift requires time. If trimming a gaze instance would shorten it below that threshold, it is instead removed. Finally, removing gaze instances may introduce gaps into the gaze sequence, where the character's gaze is unconstrained. At the beginning of such a gap, we automatically insert a new gaze instance that realigns the gaze direction with the original motion and simultaneously blends out the gaze animation (as described in Section 4.2), thus ensuring smooth transition to the original motion.

5.2 Editing Tool

We have implemented the gaze editing operations in a plugin for the Unity game engine[1]. Most of the system functionality, including gaze inference and animation playback and blending, is implemented as Unity Editor scripts. The main synthesis components—gaze controller and IK solver—are implemented as Unity components and attached to character models in the scene.

Gaze editing operations are exposed as graphical controls integrated in the Unity Editor interface. Figure 6 shows the main gaze editing controls. Gaze targets are visually indicated in the scene view (1). The current sequence of gaze instances is indicated as a directed polyline connecting the gaze targets (2). The animator can use the timeline control (3) to play or scrub along the current animation. If there is a gaze instance active at the current frame, the corresponding polyline segment is highlighted (4). Positions of the alignment handles (5) along the segments are proportional to the value of alignment parameters in the current gaze annotation (0-1 range); the animator can edit alignment values by clicking and dragging the handles. Buttons below the timeline (6) invoke gaze editing operations, such as adding a gaze instance starting at the current frame, or removing the currently active gaze instance. When the animator is satisfied with the edits they have made, they can click the Apply button (7) to synthesize the new gaze animation.

Gaze targets are implemented as tagged, dummy objects parented to scene objects and visually indicated in the scene view (Figure 6, 8). For example, there is a gaze target object attached to the gem

[1] http://unity3d.com/

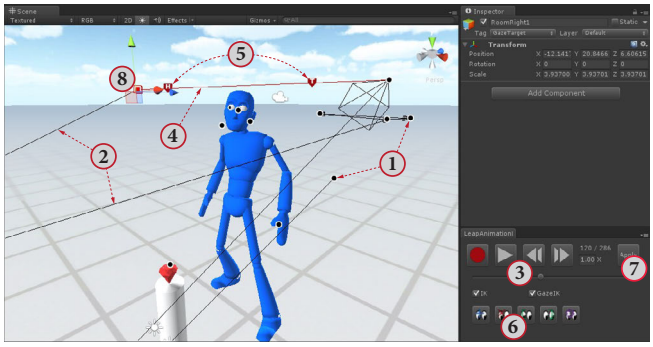

Figure 6: *Screenshot of the gaze editing interface: (1) gaze targets; (2) gaze sequence polyline; (3) timeline controls; (4) currently active gaze instance; (5) alignment parameter handles (H: head alignment; T: torso alignment); (6) gaze editing operations; (7) the "Apply" button, which synthesizes the edited animation; and (8) widget for moving a selected object or gaze target.*

in Figure 6. A large object can have multiple targets attached at different locations. Spatial editing of gaze can be done either by directly selecting and moving gaze target handles or by moving the scene objects to which they are attached. The latter method allows large-scale edits—all gaze shifts toward affected targets will adapt to their new locations.

5.3 Results

Our gaze authoring approach can produce believable results while requiring less labor and skill than traditional methods. An experienced animator must set many keyframes to add eye movements to a body motion, while our system can add an equivalent amount of animation automatically. Manually adding a new gaze shift to an existing gaze animation requires setting multiple keyframes on each of the eye, head, and torso joints, whereas our tool can accomplish the same result in a single operation. Much of the challenge in producing high-quality animation comes from getting the timing right—each keyframe needs to be placed with great care. Our tool greatly reduces this burden—the gaze controller implements kinematics of human gaze and automatically computes relative timings of eye, head, and torso movements that result in natural motion.

Figure 7 shows several examples of gaze animation created using our approach, excerpted from the supplemental video. Each example consists of two rows of images showing two versions of the same scene. Example 1 (WalkCones) demonstrates the effects of adding inferred eye movements to a walking motion. The eye movements make the character more lifelike and anticipate changes in trajectory. Example 2 (DrinkSoda) shows a story edit where a new character is introduced and the other two characters are made to gaze at her. In Example 3 (MakeSandwich), the character making a sandwich is made to look at the camera after adding each ingredient to increase the viewer's engagement. In Example 4 (WaitForBus), we prevented the character from checking his watch by removing the corresponding gaze instance and constraining his arm. Examples 5 and 6 show how we can use gaze edits to change the character's personality. Example 5 (ChatWithFriend) contrasts two versions of a first-person conversational scene, one where the character unsettlingly stares at the camera and another where it looks away uncomfortably. In Example 6 (HandShake), we made the character appear standoffish by having him look away during the handshake.

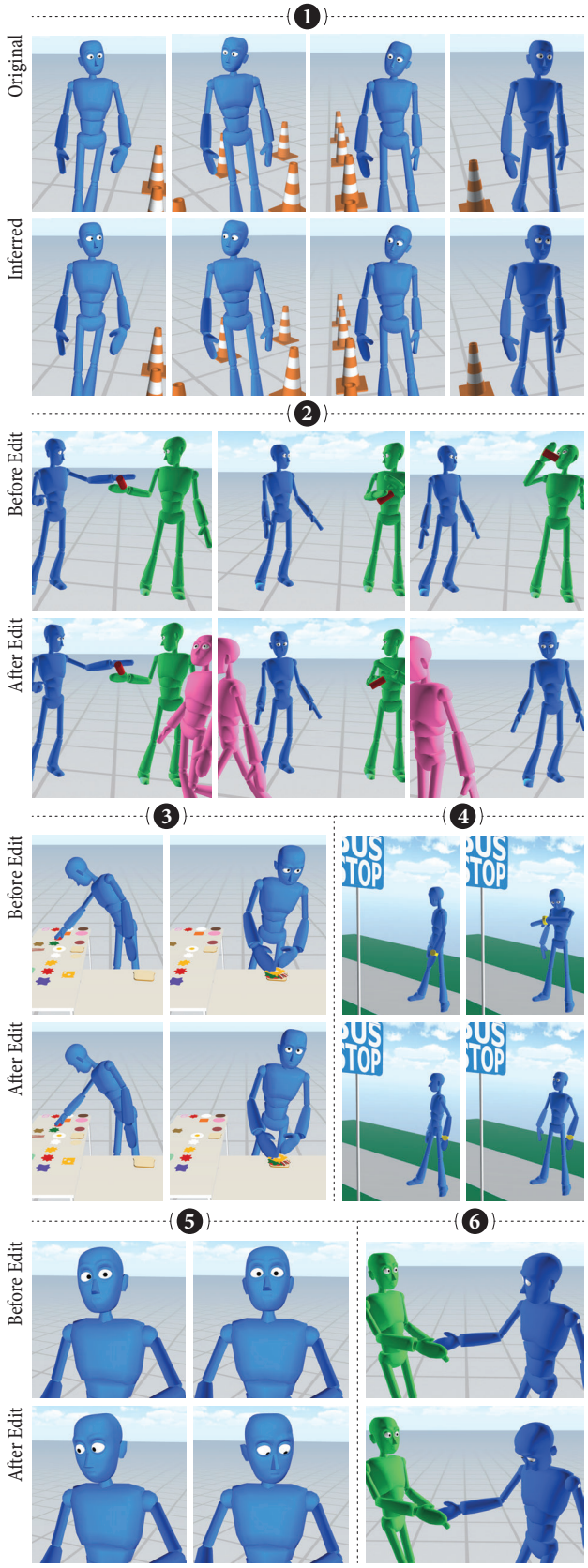

Figure 7: *Examples of gaze animation created using our system. More examples are provided in the supplemental video.*

Scene	Traditional		Our Approach	
	Time	# keys	Time	# ops
HandShake	10:40	19	3:11	8
StealGem	1:29	9	1:30	4
DrinkSoda	9:05	43	3:51	9
MakeSandwich	10:43	33	4:05	13

Table 2: *Gaze editing effort: traditional vs. our approach.*

6 Evaluation

The goal of this work is to reduce the effort required to produce high-quality gaze animation by modeling the gaze behavior as an automatically inferrable, easily editable sequence of gaze instances. We conducted a two-part evaluation of our approach. The first part assessed the amount of effort required to author gaze animation to confirm that our approach saves animator time and effort. The second part assessed whether this reduction in effort came at the expense of animation quality.

6.1 Authoring Effort

We evaluated authoring effort in two ways. First, we investigated the effort required to add eye animation to a captured body motion without modifying the motion itself, which our tool can do automatically. We asked two experienced animators to add eye movements to seven scenes, originally captured for the evaluation of the gaze inference approach (Section 3.4), with a cumulative length of 2:36 minutes. Both animators used Autodesk MotionBuilder, where they set up a simple eye animation rig that they keyed to create the desired eye movements. We measured authoring effort using two metrics: (1) time taken and (2) number of keys set. We found that the task required on average 25 minutes and 86 keys for each minute of animation. By contrast, our system can synthesize an equivalent amount of eye animation automatically, requiring on average 1.50 minutes of computation per minute of animation. This does not include the time needed to manually label important objects in the scene, which typically takes less than a minute. We note that the labeling effort scales much better with scene complexity than does specifying the target of each gaze shift directly, because objects are often gazed at multiple times. Moreover, large objects—such as other characters—often have multiple parts that get looked at, and they require multiple target handles as a result.

Second, we assessed the effort required to edit gaze animation in a scene that already contained eye movements. An experienced animator was instructed to make the following edits: (1) introducing gaze aversion in a two-character interaction scene (HandShake), (2) removing gaze toward an object in a scene that involved interaction with the environment (StealGem), (3) introducing gaze toward a new character in a multi-character interaction scene (DrinkSoda), and (4) introducing gaze toward the camera in a scene that involved interaction with the environment (MakeSandwich). All of the edits involved changing not only eye movements, but also head and (in some cases) torso posture. To accomplish the edits, the animator used layering and keyframing features in MotionBuilder. Additionally, a novice animator authored the same edits using our gaze-editing tool. As before, we measured the time taken and number of operations required. For MotionBuilder, the latter is equivalent to the number of keys set, while for our tool it refers to the editing operations discussed in Section 5. Our measurements, reported in Table 2, indicate that our approach requires 1/3 of the time and number of operations required to achieve equivalent edits using traditional motion editing.

6.2 Animation Quality

To evaluate the effect of our authoring approach on perceived quality of produced animation, we conducted a human-subjects experiment. We compared our approach to alternative methods for animating character gaze, including no eye animation, eye movements recorded using an eye tracker, and hand-authored gaze. We showed participants pairs of videos of a motion-captured character performing various actions and asked them to choose the one they thought had superior animation. The videos in each pair were identical in every way except the gaze animation method. Different methods corresponded to the following study conditions: (1) *no gaze*, (2) *recorded gaze* (using an eye tracker), (3) *hand-authored gaze*, and (4) *synthesized gaze* (our approach).

Hypotheses – We hypothesized that (1) *synthesized gaze* would be preferred over *no gaze*; (2) *synthesized gaze* would be preferred over *recorded gaze*; and (3) *synthesized gaze* would be seen as non-inferior to *hand-authored gaze*. Hypothesis 2 is motivated by the consideration that raw eye-tracking data is noisy, non-trivial to map to an animated character, and often incorrect with respect to scene requirements. Hypothesis 3 reflects the expectation that our approach can reduce authoring effort without compromising quality.

Design – The experiment involved three separate studies, each of which tested one of the hypotheses. Each study consisted of five task trials and followed a within-participants design, wherein the participants were asked to choose between videos in *synthesized gaze* and one of the other conditions. Video pairs were presented to the participants in a randomized order.

Stimuli – Experiment stimuli included 5×4 video clips (five scenes, each in four conditions) of a character animated using motion capture. The animations were 9 to 14 seconds in length. We created the *no gaze* condition videos by extracting segments from original scenes created for the evaluation of our gaze inference method (Section 3.4.) The scenes included ChatWithFriend, MakeSandwich, StackBoxes, StealGem, and WalkCones. The videos in the *hand-authored gaze* condition were created by extracting the same segments from the scenes created for our evaluation of authoring effort (Section 6.1). The videos in the *synthesized gaze* condition resulted from applying gaze inference to the segments. To create the *recorded gaze* condition, we generated eye animations directly from eye-tracking data.

Measures – Our experiment had the following subjective measures: (1) perceived *animator competence*, (2) *realism*, and (3) *communicative clarity*. The measures were implemented as binary-choice questions presented at each task trial as follows: (1) "On which video did the animator do a better job?" (2) "In which video were the character's movements more realistic?" (3) "In which video did the character communicate more clearly what they were doing?"

Participants – We conducted the experiment online using the Amazon Mechanical Turk crowdsourcing service. We recruited 24 participants for each study—a total of 72 participants. The participants were paid at the rate of $6/hour.

Results – Upon collecting our data, we conducted separate analyses for each study of the experiment. We aggregated all participants' binary answers across all the scenes and obtained, for each condition, the count of how many times it was chosen over its counterpart in each measure. Then we compared the counts using chi-square tests of goodness-of-fit. The results are described in the following paragraphs and shown in Figure 8. (*) marks significant differences.

Our analysis of data from study one found a significant preference for *synthesized gaze* over *no gaze* on all measures, $\chi^2(1,120) = 6.41, p = .011^*$ (animator competence), $\chi^2(1,120) = 3.99, p = .046^*$ (realism), and $\chi^2(1,120) = 5.54, p = .019^*$ (communicative clarity). These results support Hypothesis 1.

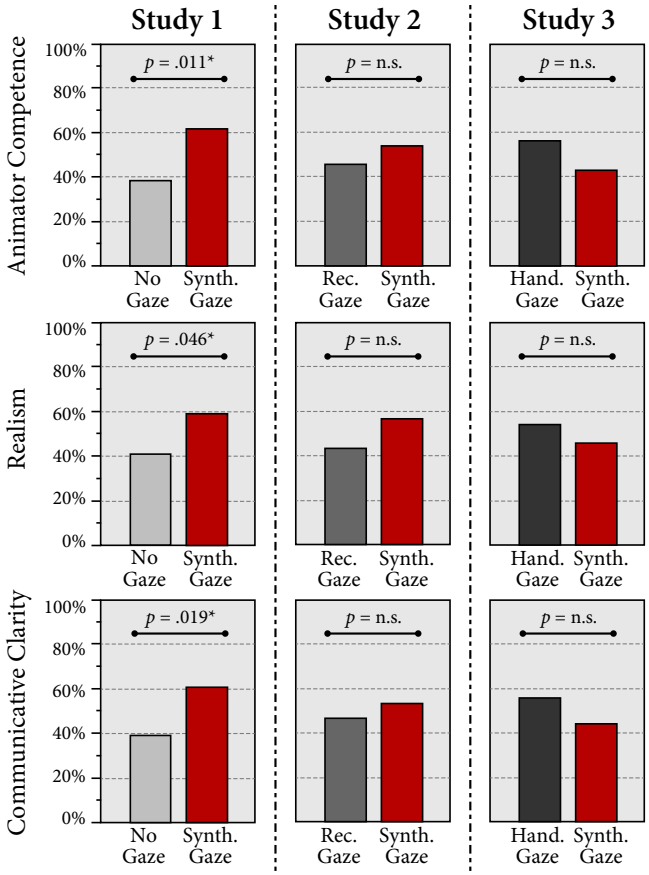

Figure 8: *Results of the study comparing gaze animation produced using our approach (synthesized Gaze) to (1) no gaze animation, (2) gaze recorded using an eye tracker, and (3) hand-authored gaze. The y-axis shows the percentage of trials in which one condition was chosen over the other.*

Data from the second study showed that *synthesized gaze* was not significantly preferred over *recorded gaze* in any of the measures, $\chi^2(1,120) = 0.83, p = .362$ (animator competence), $\chi^2(1,120) = 2.12, p = .145$ (realism), and $\chi^2(1,120) = 0.53, p = .466$ (communicative clarity). These results do not support Hypothesis 2.

In study three, we tested Hypothesis 3 using the Wald test of non-inferiority with the margin σ set to 0.07, following guidelines from literature [Ng 2001]. *Synthesized gaze* was not seen as non-inferior to *hand-authored gaze* on either animator competence ($\chi^2(1,120) = 0.000, p = 1.000$), realism ($\chi^2(1,120) = 0.048, p = .827$), or communicative clarity ($\chi^2(1,120) = 0.256, p = .513$). These results do not provide support for Hypothesis 3.

In a post-hoc exploratory analysis, we investigated the potential reasons for the lack of support for Hypothesis 2, as the findings contrasted our qualitative observations. This analysis focused on how recorded and synthesized gaze differed across different scenarios in order to determine whether or not our approach had a differential effect on different scenes. We found that scene type had a significant effect on participants' choice of *recorded gaze* in all measures, e.g., $\chi^2(4,55) = 25.04, p < .0001^*$ for the "animator competence" measure. Further inspection showed that scenes with a close-up of the

character benefited the most from our synthesized gaze approach, likely due to greater saliency of gaze cues. The differences between approaches were indiscernible in scenes where the character's eyes were small on the screen and other motion components dominated. An illustrative example is the ChatWithFriend scene, which was always chosen in *synthesized gaze* condition over *recorded gaze* in animator competence and realism measures, and 23 times out of 24 in communicative clarity.

Study results suggest that adding eye animation using our approach leads to improvement in the perceived quality of the animation over having no eye animation at all. Our approach also seems to afford benefits over recorded eye motion data in scenes where gaze behavior is perceptually dominant. However, evidence suggests that the use of our approach incurs a small loss of quality compared to expert-crafted animation.

7 Limitations and Future Work

Our gaze authoring approach could be extended in a number of ways. First, we focus on modeling only directed gaze. Although other human gaze movements such as smooth pursuit, saccades, and conversational aversions are not explicitly simulated in our system, they can be reasonably approximated as directed gaze. Others, such as eye blinks, co-occur with directed gaze and are important for expressiveness. In the current implementation of our approach, we add probabilistically generated eye blinks and saccades to the synthesis output in a post-processing step in order to achieve more expressive results. Blinks are generated following the model proposed by Peters et al. [2010], and saccadic gaze is based on the Eyes Alive model [Lee et al. 2002]. In future work, we plan to explore how such movements could be supported in a more robust fashion.

Second, while our gaze inference approach is sufficient to obtain a plausible gaze behavior that can serve as a starting point for editing, its accuracy and sensitivity could be improved further. Probabilistic formulations used for gaze-event and -target inference can be extended in many ways; for example, heuristic terms can be replaced by priors learned from ground-truth data. Saliency-based methods for idle gaze (e.g., [Peters and O'Sullivan 2003]) could be utilized to refine the output of target inference. Recorded eye movements—if available—could be utilized for the same purpose. Support for tasks that involve object interactions could be enhanced by introducing target inference terms for other significant kinematic events besides hand contact start [Johansson et al. 2001]. Moreover, the gaze-inference approach need not be restricted to non-interactive scenarios and could be adapted to operate on interactive inputs, allowing its use for predicting user attention and intent in virtual reality and other interactive contexts.

The current implementation of the system in Unity serves as an adequate proof of concept, but its utility to artists would be increased by integration with commonly used 3D-animation software such as MotionBuilder or Maya. Furthermore, the system does not provide truly interactive editing due to the limitations of the synthesis approach. While our gaze controller generates biologically plausible movements, its feed-forward design and parametrization based on relative head and torso alignments necessitate re-synthesis of the entire motion from the beginning to the end to obtain the exact editing result. To achieve interactive performance, alternative methods for motion synthesis must be explored.

8 Conclusions

We have proposed an approach for adding editable, directed gaze animation to characters animated using full-body motion capture. The approach is based on the idea of modeling the gaze behavior as a sequence of gaze instances, representing gaze shifts toward targets in the scene. We have shown how this representation can be automatically inferred from captured body motion and scene geometry, producing an initial gaze behavior that can be refined further through manual editing. We have also described a convenient approach for gaze editing that takes advantage of two key properties of this representation: abstraction of gaze pose and timing and anchoring the gaze behavior in the scene via target handles. Finally, we have described a method for synthesis of gaze motion from this representation and adding it to the original motion. As shown in our evaluations, our approach can substantially reduce the effort and skill required to author gaze animation compared to traditional tools. A novice animator can use it to endow characters with gaze at an acceptable level of quality, while skilled animators can save effort by using the output of our approach as a first-guess animation they can edit further. In particular, our approach has potential applications in domains where quantity and speed of animation production are important considerations, such as television animation or animation of background and mid-ground characters in film and games.

Acknowledgements

We thank Bobby Duncanson, Kevin Bennett, Elliot Hwang, and Jessica Nelson from Madison Media Institute for their help in obtaining test motions, as well as Arissa Sato for her annotation services. This research was supported by the University of Wisconsin–Madison Graduate School and National Science Foundation awards 1017952 and 1208632.

References

BAI, Y., SIU, K., AND LIU, C. K. 2012. Synthesis of concurrent object manipulation tasks. *ACM Transactions on Graphics 31*, 6 (Nov.), 156:1–156:9.

CAFARO, A., GAITO, R., AND VILHJÁLMSSON, H. 2009. Animating idle gaze in public places. In *Proceedings of 9th International Conference on Intelligent Virtual Agents*, IVA '09, Springer, 250–256.

COLEMAN, P., BIBLIOWICZ, J., SINGH, K., AND GLEICHER, M. 2008. Staggered poses: A character motion representation for detail-preserving editing of pose and coordinated timing. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '08, 137–146.

DENG, Z., LEWIS, J. P., AND NEUMANN, U. 2005. Automated eye motion using texture synthesis. *IEEE Computer Graphics & Applications*, 24–30.

ELKOURA, G., AND SINGH, K. 2003. Handrix: Animating the human hand. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '03, 110–119.

FISHER, R. A. 1925. *Statistical Methods for Research Workers*. Oliver and Boyd.

FULLER, J. H. 1992. Head movement propensity. *Experimental Brain Research 92*, 1, 152–164.

GLEICHER, M. 2001. Motion path editing. In *Proceedings of the 2001 Symposium on Interactive 3D Graphics*, ACM, New York, NY, USA, I3D '01, 195–202.

GRILLON, H., AND THALMANN, D. 2009. Simulating gaze attention behaviors for crowds. *Computer Animation and Virtual Worlds 20*, 23 (June), 111–119.

HECK, R. M. 2007. *Automated authoring of quality human motion for interactive environments*. PhD thesis, University of Wisconsin–Madison.

HENDERSON, J. M. 2003. Human gaze control during real-world scene perception. *Trends in Cognitive Sciences 7*, 11, 498–504.

HIETANEN, J. K. 1999. Does your gaze direction and head orientation shift my visual attention? *Neuroreport 10*, 16, 3443.

HO, E. S. L., KOMURA, T., AND TAI, C.-L. 2010. Spatial relationship preserving character motion adaptation. *ACM Transactions on Graphics 29*, 4, 33:1–33:8.

HUANG, Y., AND KALLMANN, M. 2016. Planning motions and placements for virtual demonstrators. *IEEE Transactions on Visualization and Computer Graphics 22*, 5 (May), 1568–1579.

JOHANSSON, R. S., WESTLIN, G., BÄCKSTRÖM, A., AND FLANAGAN, J. R. 2001. Eye-hand coordination in object manipulation. *The Journal of Neuroscience 21*, 17, 6917–6932.

JÖRG, S., HODGINS, J., AND SAFONOVA, A. 2012. Data-driven finger motion synthesis for gesturing characters. *ACM Transactions on Graphics 31*, 6, 189:1–189:7.

KHULLAR, S. C., AND BADLER, N. I. 2001. Where to look? automating attending behaviors of virtual human characters. *Autonomous Agents and Multi-Agent Systems 4*, 1, 9–23.

KOKKINARA, E., OYEKOYA, O., AND STEED, A. 2011. Modelling selective visual attention for autonomous virtual characters. *Computer Animation and Virtual Worlds 22*, 4, 361–369.

LANCE, B. J., AND MARSELLA, S. C. 2010. The expressive gaze model: Using gaze to express emotion. *IEEE Computer Graphics & Applications 30*, 4, 62–73.

LEE, S. P., BADLER, J. B., AND BADLER, N. I. 2002. Eyes alive. In *ACM Transactions on Graphics*, vol. 21, 637–644.

LEE, J., MARSELLA, S., TRAUM, D., GRATCH, J., AND LANCE, B. 2007. The rickel gaze model: A window on the mind of a virtual human. In *Proceedings of 7th International Conference on Intelligent Virtual Agents*, IVA '07, Springer, 296–303.

MCCLUSKEY, M. K., AND CULLEN, K. E. 2007. Eye, head, and body coordination during large gaze shifts in rhesus monkeys: Movement kinematics and the influence of posture. *Journal of Neurophysiology 97*, 4, 2976–2991.

MITAKE, H., HASEGAWA, S., KOIKE, Y., AND SATO, M. 2007. Reactive virtual human with bottom-up and top-down visual attention for gaze generation in realtime interactions. In *Proceedings of 2007 IEEE Virtual Reality Conference*, 211–214.

NG, T.-H. 2001. Choice of delta in equivalence testing. *Drug Information Journal 35*, 4, 1517–1527.

PEJSA, T., ANDRIST, S., GLEICHER, M., AND MUTLU, B. 2015. Gaze and attention management for embodied conversational agents. *ACM Transactions on Interactive Intelligent Systems 5*, 1, 3:1–3:34.

PETERS, R. J., AND ITTI, L. 2008. Applying computational tools to predict gaze direction in interactive visual environments. *ACM Transactions on Applied Perception 5*, 2 (May), 9:1–9:19.

PETERS, C., AND O'SULLIVAN, C. 2003. Bottom-up visual attention for virtual human animation. In *Proceedings of the 16th International Conference on Computer Animation and Social Agents*, CASA '03, IEEE, 111–117.

PETERS, C., AND QURESHI, A. 2010. A head movement propensity model for animating gaze shifts and blinks of virtual characters. *Computers & Graphics 34*, 6, 677–687.

POPOVIĆ, Z., AND WITKIN, A. 1999. Physically based motion transformation. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, SIGGRAPH '99, 11–20.

RUHLAND, K., PETERS, C. E., ANDRIST, S., BADLER, J. B., BADLER, N. I., GLEICHER, M., MUTLU, B., AND MCDONNELL, R. 2015. A review of eye gaze in virtual agents, social robotics and hci: Behaviour generation, user interaction and perception. *Computer Graphics Forum*.

SHIN, H. J., LEE, J., SHIN, S. Y., AND GLEICHER, M. 2001. Computer puppetry: An importance-based approach. *ACM Transactions on Graphics 20*, 2 (Apr.), 67–94.

THIEBAUX, M., LANCE, B., AND MARSELLA, S. 2009. Real-time expressive gaze animation for virtual humans. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, AAMAS '09, 321–328.

UEMURA, T., ARAI, Y., AND SHIMAZAKI, C. 1980. Eye-head coordination during lateral gaze in normal subjects. *Acta Oto-Laryngologica 90*, 3–4, 191–198.

YE, Y., AND LIU, C. K. 2012. Synthesis of detailed hand manipulations using contact sampling. *ACM Transactions on Graphics 31*, 4, 41:1–41:10.