An Actuated Physical Puppet as an Input Device for Controlling a Digital Manikin

Wataru Yoshizaki[1,2], Yuta Sugiura[3,4], Albert C Chiou[4], Sunao Hashimoto[4], Masahiko Inami[3,4], Takeo Igarashi[4,5], Yoshiaki Akazawa[2], Katsuaki Kawachi[2], Satoshi Kagami[2], Masaaki Mochimaru[2]

[1]NAIST, Nara, Japan, wataru-y@is.naist.jp

[2]DHRC AIST, Tokyo, Japan, {y-akazawa, k.kawachi, s.kagami, m-mochimaru}@aist.go.jp

[3]Keio Univ., Kanagawa, Japan, inami@inami.info

[4]JST ERATO, Tokyo, Japan, hashimoto@designinterface.jp

[5]The Univ. of Tokyo, Tokyo, Japan, takeo@acm.org

ABSTRACT

We present an actuated handheld puppet system for controlling the posture of a virtual character. Physical puppet devices have been used in the past to intuitively control character posture. In our research, an actuator is added to each joint of such an input device to provide physical feedback to the user. This enhancement offers many benefits. First, the user can upload pre-defined postures to the device to save time. Second, the system is capable of dynamically adjusting joint stiffness to counteract gravity, while allowing control to be maintained with relatively little force. Third, the system supports natural human body behaviors, such as whole-body reaching and joint coupling. This paper describes the user interface and implementation of the proposed technique and reports the results of expert evaluation. We also conducted two user studies to evaluate the effectiveness of our method.

ACM Classification: I.3.6 [Computer Graphics]: Methodology and Techniques - Interaction Techniques; I.3.1 [Computer Graphics]: Hardware Architecture - Input devices.

General terms: Design, Human Factors

Keywords: input device, posture, force feedback, robot.

INTRODUCTION

A physical prop can be a powerful input device for controlling the posture of a rigged character [1, 15, 16, 23, 9, 10]. It offers many advantages compared with standard pointing devices such as the mouse. It provides a natural three-dimensional (3D) perspective, rather than just a two-dimensional (2D) screen, and both hands can be used to control 6 × 2 degrees of freedom (DOFs) without difficulty [13]. Moreover, the physical prop can provide direct tactile feedback, which is missing from abstract 3D input devices. Most importantly, the system can leverage a person's built in "hand intelligence" in operational tasks. However, existing prop-based input devices are mostly passive. The user sets the individual joint angles manually, but the system does not provide any active feedback. Also, these devices must usually be held by the user to maintain the desired posture against the force of gravity.

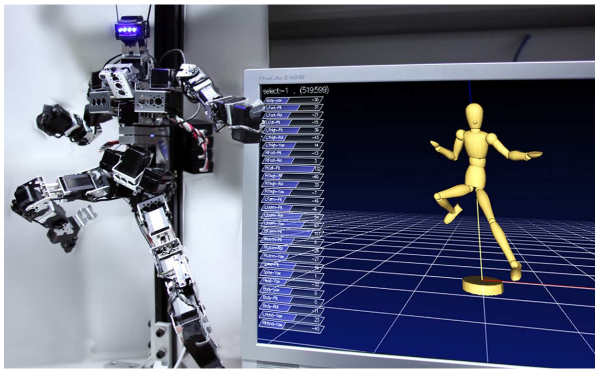


Figure 1: Active marionette system. The user controls the character by manipulating the device.

In this research, we add active control to a prop-based input device. A servomotor is attached to each DOF of a joint, and drives the joint. Actuation is useful in many ways. First, the user can upload pre-defined postures or the results of previous edits to the device to save time. Second, the system is capable of dynamically adjusting joint stiffness to counteract gravity, while allowing control to be maintained with relatively little force. Third, the system supports natural human body behaviors, such as whole-body reaching and joint coupling, by using measured human data.

Our primary target application is control of the digital or computer manikins used in product design [8,14]. A designer inserts a digital manikin in a virtual model of a product being designed in order to carry out ergonomics assessments of such factors as reachability and visibility. For example, an automotive designer places a virtual driver on the seat to ascertain that the driver can see and reach the console. Digital manikins have become a fundamental technology that is indispensable in industry. Various commercial products already exist (e.g., Jack from Siemens, RAMSIS from Human Solutions) and are widely used in the design of automobiles, airplanes, factory assembly lines, etc. However, a digital manikin has many degrees of freedom, and its control is notoriously difficult. Various techniques such as inverse kinematics have been proposed, but the problem has not yet been completely solved. We hope to address this issue by introducing a novel hardware solution and thus contribute to the manufacturing industry.

We first discuss related work and explain the basic user-interaction features of the device. We then describe the implementation of our prototype system, as well as the results of qualitative evaluations by experts and quantitative evaluations by test users. One evaluation compared our system with the standard mouse-based technique and a passive puppet device. Another compared the performance of our system with and without data-driven inverse kinematics. We conclude with a discussion of possible application scenarios for our system.

RELATED WORK

Several previous attempts have been made to use physical props as input devices for character posture control. Knep et al. [16] and Esposito et al. [9] developed systems for animating an articulated figure via a physical skeleton covered with sensors that monitor the orientations of the joints. Feng et al. [10] built a similar system with a vision sensor. Hinckley [13] introduced a prop-based, two-handed user interface for neurosurgical visualizations, allowing a surgeon to specify a cross-section for viewing by holding a plastic plate next to a miniature head. Weller et al. [23] introduced a hub-and-strut construction kit called Posey, in which the components report their physical configuration and relative orientation to a computer, which then constructs a corresponding virtual representation. Although these solutions have overcome a number of difficulties inherent in working with a 2D environment, the input devices are passive and thus can only pass information to the virtual environment without providing any feedback or active support to the user.

Force feedback is already popular in virtual reality applications such as surgical simulations [4] and remote manipulator control [22]. Some systems use force feedback for geometric editing [11] and painting [2], but their goal is to provide specific sensations to the user's finger or hand; the form factor of the device itself is not significant in these applications. In contrast, the shape of the input device is critical in character posing, and force feedback is used to assist in the control of a rigged character.

Although most of the research effort in robotics is focused on autonomous behavior, some systems do explicitly focus on interactivity. Sekiguchi et al. [20] developed the RobotPHONE system, in which two teddy bear robots are remotely connected through a network, and their postures are synchronized. Raffle et al. [18] created Topobo, a 3D constructive assembly system with kinetic memory, in which the actuator component recalls user operations and replays the motions. Shimizu et al. [21] introduced a teddy bear robot as a generic game controller. The main applications of these systems are entertainment and education. Our work builds on these attempts and uses a robot for the control of digital manikin with more DOFs in more goal-oriented tasks.

ACTUATED PUPPET INTERACTION

This section explains the features of our active puppet device from the user's point of view. Implementation details will be described in the next section. The basic idea is to add a servomotor to each joint of a puppet input device to support posture control by providing active feedback. In this work, we focus on the following three features: posture loading, gravity compensation, and active guidance using measured human data.

Posture loading

The posture design process rarely starts from scratch and often involves a small modification of an existing posture. The user might begin with a predefined canonical posture, such as sitting, standing, crouching, etc., and then adjust it to obtain a specific target posture. Alternatively, the user might start with the result of a previous editing task and further explore that result with additional modifications. Animation consists of a lengthy sequence of slightly different postures, and the design of a posture in a given frame usually begins with the posture in the previous frame.

Active joints can support this process by loading the existing posture into the puppet device. The user first selects the initial posture on the computer screen, loads the posture into the puppet device, and then modifies the posture by manipulating the puppet. This process is much simpler and faster than the use of a passive puppet, whose posture must be manually matched with the initial posture.

Intelligent Gravitational Compensation

Most existing passive puppet devices cannot independently maintain their postures against the force of gravity and must be held by the user. If each joint were stiff enough to counteract gravity on its own, the joints would be too hard to rotate manually.

Active joints address this problem by dynamically changing their stiffness. If the user is not controlling a joint, the system makes the joint stiff enough to hold its configuration against gravity. If the user is controlling the joint, the system makes the joint flexible enough to be freely rotated by the user.

Our current prototype puppet device successfully maintains its posture against the force of gravity, while allowing the rotation of a joint with a force as small as that typically applied by the little finger.

Active Guidance Using Measured Human Data

The human body has many hard and soft constraints, and it is important to design postures satisfying these constraints. Otherwise, the results will look very unrealistic. An active puppet can assist the user in complying with these constraints by providing active feedback based on measured human data.

One important point is that these constraints are not independent for each joint. If the constraints were simply a matter of an acceptable range of rotation angles for each joint, it would be straightforward to implement them with a passive puppet. However, some constraints involve multiple joints, and enforcement of these constraints is only possible with active feedback.

Our current prototype system supports natural reaching and joint coupling by using measured human data. Natural reaching takes the position of a hand as input and adjusts the joint angles to bring the hand to the desired position. The human body has redundant degrees of freedom for bringing an end effecter (e.g., a hand) to a position in space, and measured data resolve this ambiguity. Joint coupling is the interdependency between nearby joints, and our current prototype supports coupling between hip and knee joints. As the angle of the hip joint approaches 90 degrees, it becomes difficult to keep the knee joint straight.

IMPLEMENTATION

Our active puppet system consists of a handheld puppet device and a control program running on a host PC (Figure 2). All joints of the puppet device are actuated by servomotors made for hobby purposes by Futaba Co., with an RS485 serial interface to control and measure the parameters. The servomotors are connected to the host PC via cable. The host PC uses a simulation model of the puppet for physical simulations, and the postures of the puppet and the simulation model are synchronized at up to 15 Hz. A human interface program displays a 3D graphical model of the virtual puppet. This visualization model may have a different shape from the handheld puppet.

When the operator changes a joint angle of the handheld puppet, the system changes the corresponding joint angle of the simulation model, and this is reflected in the visualization model shown on the display via a mapping function. When the operator changes a joint angle of the visualization model, the change is passed to the simulation model, and the system changes the corresponding joint angle of the puppet device by controlling the appropriate servomotor.

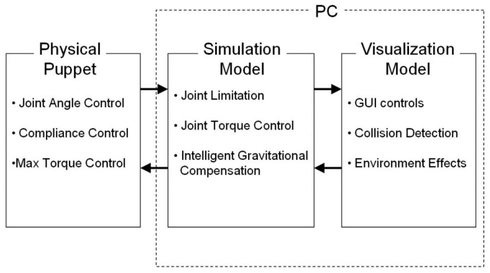
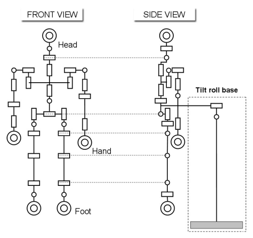


Figure 2: System configuration.

Handheld Puppet Hardware

The puppet is 40 cm in height and has 32 DOFs (joints). Figure 3 shows the overall configuration and joint specifications. The head and torso have 3 DOFs each to achieve human-like motion, such as standing on one leg. The shoulder has an extra degree of freedom to pull the arm to the front, allowing natural arm motion. The knee has a large joint angle to accommodate the "Seiza" posture (sitting down Japanese style, with the buttocks on top of the ankles).

Additionally, there is a hook for hanging the puppet by the waist joint to help generate free leg motion. With this, the user does not need to support the device manually. The hook has two DOFs to measure and control the roll and pitch of the entire body. (whole body part in Figure 3)



Body Part	Joint	Axis Direction	Servo
Arm	Shoulder	Roll, Yaw	RS302CD
		Pitch	RS301CR
	Elbow	Pitch	RS302CD
	Wrist	Yaw	
	Clavicle	Yaw	
Leg	Ankle	Pitch, Roll	RS301CR
	Knee	Pitch	
	Hip	Roll, Pitch, Yaw	
Torso	Waist	Roll, Pitch, Yaw	
Head	Neck	Roll, Pitch, Yaw	RS302CD
Whole body	Spine	Roll, Pitch	RS601CR

Figure 3: Hardware configuration.

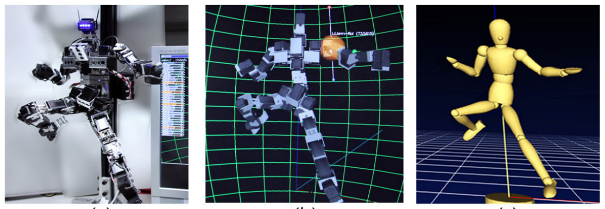


Figure 4: (a) puppet, (b) simulation model, (c) visualization model.

Communication between the Puppet and the Host PC

The servomotors (RS301CR, RS302CD, RS601CR, Futaba Co.) used for the puppet have RS485 serial communication to measure and control each joint. Table 1 shows their specifications. The parameters of these servomotors are target joint angle, servo gain, maximum torque, current joint angle, current torque, current temperature, and so on. The servo motors do not need any calibration. The servomotors are daisy chained and conduct two-way communications with the host PC using the RS-485 protocol. The host PC controls the target angle, servo gain, torque on/off, and maximum torque of each servomotor. The servo gain defines the relationship between the displacement angle and torque with which the system controls the stiffness of each joint. Each servomotor sends the current angle and temperature back to the host PC. The system then estimates the current torque from the current angle and servo gain setting (see Appendix for detail).

We use 115,200-bps serial communications in our current implementation, and the main control loop (which reads the data from the servomotors, computes the necessary torques, and sends the control commands to the servomotors) operates at approximately 10 fps. Hence, there is a delay of approximately 100 milliseconds before the user receives the force feedback from the system. This rate is significantly slower than standard haptic devices, but it is adequate because our goal is to assist the user's modeling operations, rather than to provide sensory illusions. Our current target is the design of static postures, and so this rate is acceptable. However, it might be necessary to support a faster loop for the design of animations by real-time performance.

Table 1 :Servomotor Specifications

	RS301	RS302	RS601
Torque	5	7.1	21
Baud rate [kbps]	9.6 ~ 460	9.6 ~ 460	9.6 ~ 1300
Weight [g]	21	27	91
Size [mm]	35.8 × 19.6 × 25.0	35.8 × 19.6 × 25.0	59 × 26 × 47.1
Speed [sec/60deg]	0.11	0.16	0.17
Angle Range [deg]	-150 ~ 150	-150 ~ 150	-120 ~ 120
Power [mA]	110	100	150

Simulation Model and Visualization Model

The simulation model maintains the configuration of the puppet device and its physical properties, such as the center of gravity and weight of each limb. The system matches the posture of the simulation model to that of the puppet device and uses the simulation model to predict the force that must be applied to each joint for gravitational compensation.

The visualization model represents the target character of the modeling task. The body configuration of the visualization model is freely adjustable. The mapping between the postures of the simulation model and the visualization model can be chosen arbitrarily, but in our current implementation we directly map raw joint angles. This can cause a mismatch between the postures of the device and the visualization model. For example, a posture that is possible for the visualization model may not be possible for the puppet device because of collisions, and vice versa.

However, the mismatch is usually not very large, and the user can obtain a reasonable posture with the puppet device. If the posture of the visualization model requires fine-tuning, the device can be controlled while looking at the visualization model on the screen, as with mouse-based control. Collisions can a problem, but they do not occur frequently because our targets are mostly reaching and manipulation, rather than folding arms or crossing legs.

Intelligent Gravitational Compensation

The system must make each joint stiff enough to resist the force of gravity when it is not being controlled by the user. This is accomplished by applying a torque to each servomotor. However, a joint that is too stiff is difficult for the user to rotate. The typical approach is to apply a constant stiffness to resist gravity, but we found this to be a hindrance to quick posture control. We therefore adaptively reduce the stiffness of a joint while the user is controlling it. The detailed procedure is as follows.

The system constantly adjusts the servo gain (stiffness) by computing the necessary torque at each joint to compensate for gravity, taking into account the physical properties of the device and the current posture. This keeps the joint angles constant as long as no user intervention occurs. When the user applies force to rotate a joint, the system observes a small rotation at the joint as a difference between the current (θ) and target angle (θ_d). Using this information, the system identifies the initiation of user control and reduces the stiffness by adjusting the servo gain. This allows the user to control the joint with a smaller amount of force. When control of the joint is relinquished, the user must hold the position for a short time. The system identifies this pause by detecting no change in the angle of rotation for a certain period, and then restores the original stiffness. Our method is based on a standard gravity compensation technique in the literature [5]. We tailored a popular method to work with our hardware configuration. Details are in the appendix.

Adaptive Range of Motion

There is a hardware limitation on each joint angle. A servomotor can usually rotate about 300 (deg), but this angle is sometimes too wide compared with the actual human range of motion, especially at the knees and elbows. Maximizing the torque when a joint angle reaches a human limitation assists users in generating realistic postures.

Joint Coupling

The joints of the human body are not independent, and they interfere with one another. For example, the knee joint and the hip joint are coupled. If the hip joint is rotated to 90°, it is difficult to hold the knee joint straight. We implement this joint coupling to facilitate the design of natural human postures. Because joint coupling is most prominent at the knee and hip joints, we do not include coupling of other joints in the current system. Figure 4 shows the acceptable region of hip and knee joint combinations. When the angles are outside this region, the system applies a torque to retract the limb to the nearest point inside the region. An expert in biomechanics defined the region manually based on his domain knowledge and the human posture database.

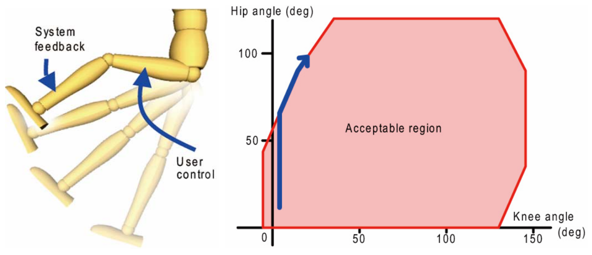


Figure 5: Joint coupling.

Natural Reaching by Data-driven Inverse-Kinematics

The active puppet system can generate natural whole body posture for a reaching action using measured human data. It calculates plausible whole body joint angles for the given hand position based on the posture database consisting of more than 1,500 different postures acquired from an optical motion capture system [24,19,12,17].

The data is obtained by measuring one subject person's pointing actions. He pointed given targets in both standing and sitting postures. The target points covered the surface of 100cm diameter cylinder around the subject's torso at 10cm grid. We captured 1536 sitting poses and 1584 standing poses.

When the user moves the puppet device, the system first computes the hand position from the current joint angles and simulation model configurations. The system then computes the desirable joint angles using the database and updates the posture of the puppet device as follows. We first compute a radial basis function (RBF) that interpolates the training data set. It takes hand position (in cylindrical polar coordinates around the body) as input and returns joint angles of the entire body as output. We then apply this RBF to the current hand position to obtain the desired whole-body pose. The hand position obtained by this interpolation does not exactly matches with the current hand position, so the system then applies standard inverse kinematics to match the hand position.

APPLICATION SCENARIOS

This section introduces several usage scenarios of the active puppet system in digital manikin applications, and reports on our informal experience with our prototype device.

Reachability assessment

One of popular applications of a digital manikin is to assess whether a specific position in a product is reachable from the user [6]. The designer places a digital manikin in the virtual scene and changes the posture of the manikin to examine if a target is reachable or not. For example, a designer places a digital manikin in front of a workbench and examines whether a tool on the bench is reachable, or places it on a car seat and determines whether control panels are reachable.

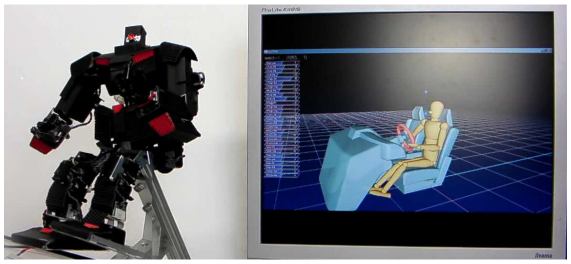


Figure 6: Reachability assessment.

We tested reachablity assessment for a car interior design scenario using our device. We prepared a 3D model of a car interior. It contains various devices that the user needs to touch and operate by hand while sitting on the seat, such as the steering wheel, driving handle, gearshift, and control panel for the air conditioner and audio system. The user places a digital manikin on the seat and reaches for these devices and tools by reaching the digital manikin's hand. If the hand touches a device, the device turns red to indicate that it is reachable. We tested this scenario with and without our active puppet device. It was much easier and more intuitive to operate using the device and we also found that the resulting postures are more natural compared to mouse control. Data-driven IK was particularly helpful because we can examine side effects of a reaching action caused by natural human body constraints such as the left arm moves as the user reaches with the right arm.

Load assessment

The other typical application of a digital manikin is to assess the load of each joint (or muscle, tendon, caput, socket, etc) under a given posture or task. If the load is too high, it causes damage to the user, so the product design needs to be changed. The active puppet system computes and displays the load of each joint in real time, so the user can easily evaluate and change the posture. The data-driven inverse kinematics further facilitates this process by automatically generating a natural posture for the given input.

We implemented an application in which the system computes the load applied to each joint from the current posture of the digital manikin. It shows joints with high load in red and those with low load in green as shown in Figure 7. We tested various postures on the application using the active puppet device and found that real-time feedback is very helpful to understand the relation between a posture and load distribution, as well as to find a posture with minimum load.

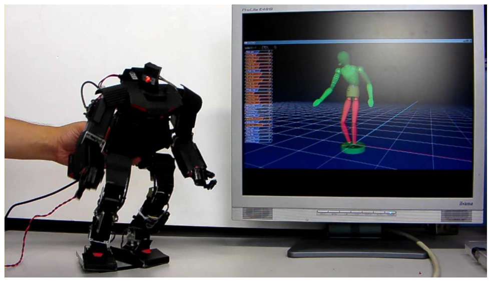


Figure 7: Load assessment

Visibility analysis

Visibility analysis examines whether specific targets are visible from the perspective of a person placed in the product. For example, the view from a car driver sitting in the driver's seat changes depending on the seat height and steering wheel position, and it is important to ensure that the driver clearly sees the outside environment. Visibility analysis is done by placing a digital manikin in a virtual environment and examining what is viewable from the digital manikin's eye position

We implemented an application in which a digital manikin sits on a driver seat and the system visualizes the visible area from the manikin's perspective. Visible area is shown as a pink cone as shown in Figure 8. Since the visible area depends on not only the head orientation but also overall posture of the character, it is beneficial to be able to control the entire body easily using the active puppet device. The active puppet device also makes it possible to control the posture while watching the manikin's view on the screen, which is very difficult for a mouse-based posture control.

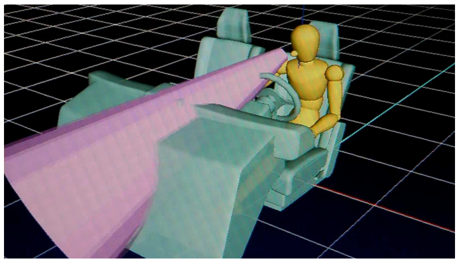


Figure 8: Visibility analysis

EVALUATION BY PROFESSIONAL USERS

We introduced our prototype system to professionals and asked them to evaluate it as potential users. We explained the concept, demonstrated the puppet system, and asked them to try it before soliciting feedback. We visited two companies. One was a game production company, and the other was a company that develops digital manikins.

Feedback from a Game Developer

We first visited the game production company and talked to a modeling expert. This individual is a member of the computer graphics department and works on geometry modeling and character posing. We introduced the device to him and then asked him to play with it for a while. The following is a summary of his feedback.

"The benefit of this device seems to be intuitive control. However, trained expert designers can fluently control character posture with a traditional user interface (manipulation of various widgets on the screen with a mouse), so the benefit of the puppet device is limited. I agree that it is nice to feel collisions with the environment as haptic feedback, but expert designers can easily design collision-avoiding postures with a traditional interface.

"A serious problem with the puppet device is that the joint structure is fixed. Game characters have various forms, and it is often necessary to handle a large variety of forms. Even human characters have a variety of configurations in a game. Furthermore, designers often assign special DOFs to control muscle bulging and flexible deformation of an arm, but these motions cannot be edited using the puppet.

"So I see limited value in this device for professional designers. However, I think the device might be useful for casual users to control character postures. For example, the puppet can be used as a controller for a character in a game. In this case, intuitive control without training is certainly attractive, and active feedback can be used to provide an engaging experience to the user. For example, the user can feel contact with the environment. It might be interesting to make the body difficult to control (stiffer) when the character is damaged in the game. "

Feedback from a Digital Manikin Developer

We then visited a company that develops various computer graphics softwares and devices. We talked to approximately 20 engineers who are working on the development of digital manikins and force feedback devices. We first introduced the device, and then asked them to play with it by themselves. The following is a summary of their comments.

"Digital manikins are used in the design of products such as furniture and car seats. The user can confirm that the size and form of the current design are ergonomically appropriate by placing the manikin in the product and checking for collisions without building a physical mock-up. A digital manikin usually provides a predefined set of body forms and postures that are transferable among different bodies. Some digital manikins have publicly available control APIs, so that control of existing commercial digital manikins is technically feasible with this puppet device.

"After playing with the device for a while, we feel that it is much easier to control character posture with the device than with a mouse. The users of digital manikins are mostly product designers and are not experts in character posing. Therefore, an intuitive control interface is very important. The ability of a device to maintain its own posture after release significantly contributes to productivity. Existing input devices such as Phantom and Polhemus cannot compensate for gravity and must be held by the user all the time. Active joints are also useful for representing various disabilities, such as paralysis on one side.

"We do see various limitations in the current implementation. First, there is a mismatch between the configuration of the puppet device and that of the digital manikin. This can cause unexpected collisions during control. The problem might be resolved to some extent by carefully tweaking the mapping, but this is not a perfect solution. It might be necessary to provide devices with different configurations or to make the puppet device physically reconfigurable. The number of joints might not be enough for some applications. We would probably need separate puppets for special purposes (for example, an arm-only device with many DOFs might be useful for detailed arm posturing). An active puppet seems to be useful for the design not only of static postures, but also of animations. It would be better to be able to use the device to represent a jumping motion by allowing vertical movement at the base."

COMPARISON WITH MOUSE AND PASSIVE PUPPET

We conducted two user studies. This section describes the first study, in which we compared our active puppet device with a mouse interface and a passive puppet device. The goal was to show that our active device with gravity compensation makes posture control easier and faster. Intelligent inverse kinematics had not been implemented at this point.

Participants

We invited 11 participants for the study. They were all male university students in computer science. Two of the participants had prior experience in character posing, whereas the other nine had no posing experience. They were all new to the puppet input device.

Methods

We compared the proposed active puppet system with a traditional mouse-based interface and a passive puppet device. We implemented a custom-designed mouse-based interface for controlling the character posture. The reason we decided not to use existing commercial programs was so that we could make all the test conditions as similar as possible. The sophisticated methods of commercial packages such as Maya and Poser have their own constraints and make it difficult to consistently use identical settings. In this study, the user controlled the joint angles either by using sliders or by dragging end effectors (such as hands and feet) with inverse kinematics. The user could also control the viewing direction by dragging the mouse. Figure 10 shows a screen shot of the mouse-based interface.

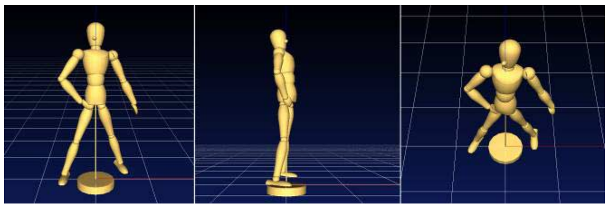


Figure 9: Example of a target posture shown to the participants.

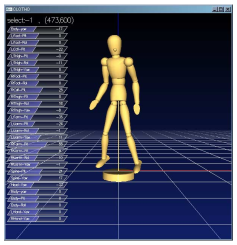


Figure 10: Screen shot of the mouse interface

The active puppet method used the system introduced in the previous section. We disabled uploading of pre-set postures because it provides too much advantage to the technique. Instead, we added a reset function so that the user could set the character posture back to a default position. We also disabled collision detection with the environment to minimize the factors influencing the results.

The passive puppet method used the same hardware as the active puppet method, but the puppet acted only as an input device. One way to implement this would be to turn off all the servomotors, but this would have been too difficult to control, as the user would have had to manually support all joints against the force of gravity. Therefore, we assigned a constant stiffness to each joint to compensate for gravity. This necessitated the use of more force to rotate a joint than with the active puppet used in this study.

Procedure

We used a within-subject procedure, in which each participant tested all three methods in a balanced order. Each participant designed three postures using each method, for a total of nine different postures in all. The same nine postures were assigned to all participants. The nine postures consisted of three groups of three postures: a group consisting of simple symmetric postures, a group consisting of asymmetric postures, and a group consisting of postures involving self-collision. The three control methods (passive, active, and mouse) were applied to these groups in a balanced way. Three screen shots of the character in the target posture, taken from different viewing angles, were printed on paper and presented to the participant, as shown in Figure 9. We imposed a cutoff time of three minutes to prevent excessively long operations. At the end of the session, we asked each participant to complete a questionnaire.

Results

Figure 11 and Figure12 show the task execution time and accuracy. We measured the accuracy as the sum of the squared errors of the joint angles. The error bar displays a 95% confidence interval. We analyzed the data via analysis of variance and found significant main effects for task execution time ($F_{2,20}$ = 33.72) and task execution accuracy ($F_{2,20}$=3.33). We then run a Tukey-Kramer post hoc pairwise comparison and the result is shown in Figure 11 and Figure12 as * marks (they show statistically significant differences ($p<0.05$)).

The active puppet was significantly faster than the mouse-based interface and the passive puppet (p < 0.05). Comments from the participants and our own observations showed that the automatic reduction of joint stiffness during control played an important role.

The active puppet was not significantly more accurate than the passive puppet (p > 0.05), but was significantly more accurate than the mouse-based interface (p < 0.05). This is somewhat counter intuitive (mouse seems to be more accurate), but the main reason is that many users failed to complete a task within the cut off time when using a mouse and this significantly increase the average errors.

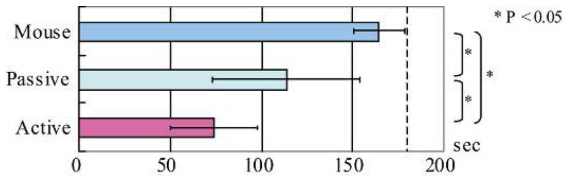


Figure 11: Task execution time.

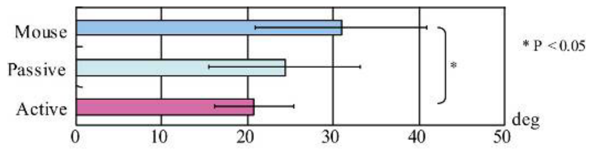


Figure12: Task execution accuracy.

Some participants complained that the end-effector positions of the virtual character were different from those of the puppet. This is because the physical structure (limb length and joint structure) of the virtual character is different from that of the physical puppet, which is a fundamental limitation of puppet devices. However, the users were still able to obtain most of the postures by controlling the individual joints while observing the posture of the virtual character on the screen. Although we did not explicitly explain this in the study, we feel that it will be necessary to include such an explanation in a tutorial for future first-time users.

Figure 13 shows the results of the questionnaire. We analyzed the data via analysis of variance and found a significant main effect on all questions ($F_{2,20}$=282.1, 301.9, 242.4, and 245). We run a non-parametric analysis (Steel-Dwass method) and the results are shown in Figure 13 as * marks (they show statistically significant differences ($p<0.05$)). Five of six participants chose the active puppet as the best method. One participant judged the active and passive puppets to be equal. This participant reported that the puppet device seemed too fragile, and he operated it very slowly, making it difficult to observe any differences between the active and passive methods.

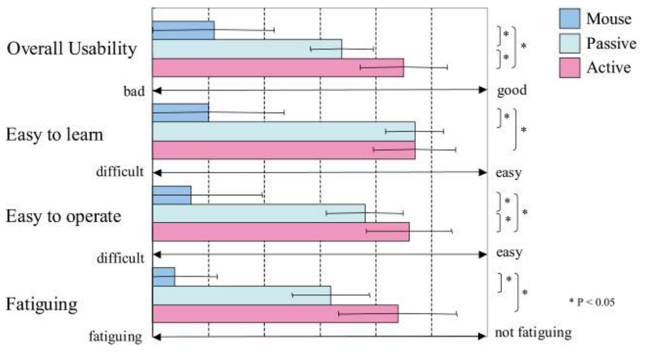


Figure 13: Questionnaire results.

The questionnaire results also showed that the active puppet caused the least amount of fatigue. We expected that the active puppet would cause less fatigue than the passive one, but it is significant that it also caused less fatigue than the mouse-based interface. Comments from the participants indicated that the active stiffness control and shorter operating time contributed to this result.

This result demonstrates the potential effectiveness of active feedback in this particular context, but it still cannot be concluded that our active puppet device is better than the best possible passive device. An ideal passive device would not have servomotors at each joint, and thus would be much lighter and easier to control. We did not test such a passive device because building special hardware solely for the purpose of this study would have been too costly. A perfectly fair comparison would be difficult to accomplish in any case if the form factors of the devices were significantly different.

EVALUATION OF DATA-DRIVEN INVERSE KINEMATICS

We ran a second user study to evaluate the effect of intelligent inverse kinematics (IK). In this evaluation, we asked the user to design reaching postures using our active puppet device with and without data-driven IK.

Participants

Ten participants joined the study. They were undergraduate or graduate students majoring in computer science. No participants except one had experience in virtual character posing using commercial software. Three users had joined the study described in the previous section.

Methods

The task was to pose a character using our active puppet device so that the hand reaches a target position in the environment. The participants designed given postures with and without data-driven IK and we compared task completion times and the quality of the resulting postures. We enabled gravity compensation in both conditions.

Procedure

We first described how to use the system for approximately 3 minutes to a participant and had him or her practice it for approximately 5 minutes. Figure 14 shows the virtual environment we used. The user points slots in the shelf highlighted in red by the right hand of the digital manikin. The digital manikin points at the top-left corner before starting a task. Starting from this configuration, the task was to point to the left-bottom, right-bottom, top-right, and top-left corner in that order. We measured the time need to complete this cycle (4 pointings). Each participant tested both conditions in a balanced order. They answered questionnaire at the end. They subjectively rated the quality of the postures they just designed in the questionnaire.

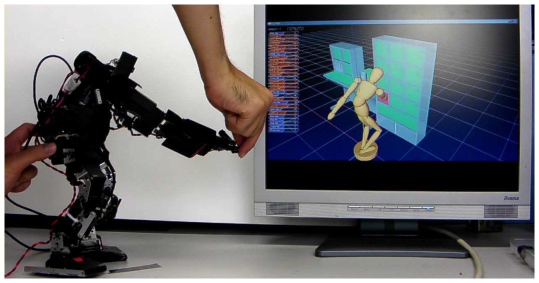


Figure 14: Environment used in the study.

Result

Figure 15 shows the task completion time. Error bars show the standard deviation. A paired t-test shows a significant difference between the two ($p < 0.05$). The graph shows that the participants completed the task approximately two times faster with data-driven IK. Figure 16 shows the results of the users' own subjective evaluation of the quality of the resulting postures (5 is the best). These results also shows that the posture designed with data-driven IK was significantly more satisfactory ($p<0.05$). We observed that the quality of the resulting postures showed large variations depending on the user's skill when designing without data-driven IK, while the quality was uniformly high when using data-driven IK.

Participants especially appreciated that data-driven IK automatically controls the lower body (waist and leg) joints when moving the hand. Control of the lower body was necessary when the user points to high or low targets. This was tedious when data-driven IK was not enabled, because the user needs to control the lower body manually.

Note that data-driven IK does not always produce an exact posture the user wants. The user may want to intentionally design a posture that is different from that in the database. We expect that the user will first posture the digital manikin approximately by using data-driven IK and then adjust the posture in detail by turning off data-driven IK.

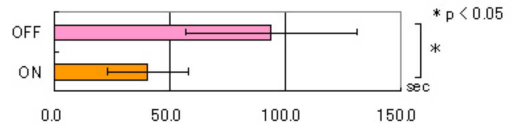


Figure 15: Task completion time

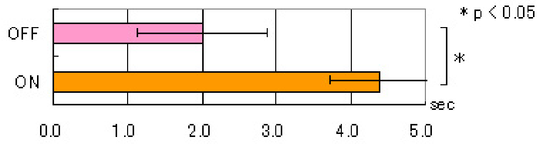


Figure 16: Subjective evaluation of the postures.

LIMITATIONS AND FUTURE WORK

One limitation of a physical puppet is that exactly matching the structure of the device to that of the target character is impossible. This results in a mismatch between the end-effector positions for the device and the virtual character. However, the user can still obtain most postures by controlling the individual joints while observing the posture of the virtual character. An exception to this would be when a collision occurs between the physical device and another object. In that case, it would be necessary to introduce some sort of clutching mechanism, or switch to a standard mouse interface to control the virtual character directly. In general we believe that the benefit of using active puppet outweigh these limitations especially in rapid prototyping phase.

Although we have focused on the design of static postures in this research, it should be possible to record dynamic motion with the device [7], and active feedback could be useful during such a procedure. For example, it might be possible for the user to swing one leg and let the system drive the other leg to design a walking motion [3]. Dynamic motion is more challenging because it requires faster actuation. However, the current hardware prototype can move reasonably fast, and we plan to explore this direction in the future.

Our current hardware implementation is an initial prototype and is built from off-the-shelf motors and frames, so the appearance is not very sophisticated. In the future, we plan to build a custom-made servo and frames to make a more human-like robot.

CONCLUSION

In this paper we proposed to add actuation to a physical puppet device and showed its feasibility with solid prototype implementation and a series of evaluations. The main technical novelty is in the data-driven inverse-kinematics applied to physical puppet. We showed that the active puppet method is faster than a mouse-based interface and the passive method with the same hardware in a study. We also showed that data-driven IK is useful for inexperienced users to obtain natural human postures. We demonstrated the device to developers of digital manikin and they saw value in rapid posture design made possible by the device.

REFERENCES

1. Arai, K. Cyber Bunraku. *ACM SIGGRAPH 97 Visual Proceedings,* (1997).
2. Baxter, B., Scheib, V., Lin, M. C., and Manocha, D. DAB: Interactive Haptic Painting with 3D Virtual Brushes. *Proc. SIGGRAPH '01*. (2001) ,461–468.
3. Chai, J., and Hodgins, J. Performance animation from low-dimensional control signals. *ACM Trans. Graph*. 24, 3 (July 2005), 686-696.
4. Cover, S. A., Ezquerra, N., O'Brien, J., Rowe, R., Gadacz, T., and Palm, E. Interactively Deformable Models for Surgery Simulation. *IEEE CG&A*, 13, 6. (1993), 68–75.
5. De Luca, A. and Panzieri, S. Learning gravity compensation in robots: Rigid arms, elastic joints, flexible links. *IJACSP*, 7, 5, (1993) 417-433.
6. Dempster, W. T. Space Requirements of the Seated Operator. *Technical Report WADC–TR–55–159*. Aerospace Medical Research Laboratory, Wright-Patterson Air Force Base, OH, (1955), 105–109.
7. Dontcheva, M., Yngve, G., Popović, Z. Layered Acting for Character Animation. *Proc. SIGGRAPH 2003*, (2003), 409–416.
8. Duffy, Vincent G. (eds.), *Handbook of Digital Human Modeling*, CRC Press, FL (2009)
9. Esposito, C., Paley, W. B., and Ong, J. Of mice and monkeys: a specialized input device for virtual body animation. *Proc. I3D '95.* (1995), 109-114.
10. Feng, T.C., Gunawardane, P., Davis, J., Jiang, B. Motion capture data retrieval using an artist's doll. *Proc. ICPR 2008*, (2008), 1-4.
11. Gaylean, T. A. and Hughes, J. F. Sculpting: An Interactive Volumetric Modeling Technique. *Proc. SIGGRAPH 1991*, (1991), 267–274.
12. Grochow, K., Martin, S. L., Hertzmann, A., and Popović, Z. Style-based Inverse Kinematics. *ACM Trans. Graph*. 23, 3 (Aug. 2004), 522–531.

13. Hinckley, K. Haptic Issues for Virtual Manipulation, Ph.D. Thesis, University of Virginia, (1997).

14. ISO 15536-2:2007(E) Ergonomics Comp. Manikins and Body Templates Part 2: Verification of Functions and Validation of Dimensions for Comp. Manikin Systems.

15. Johnson, M. P., Wilson, A., Blumberg, B., Kline, C., and Bobick, A. Sympathetic Interfaces: Using a Plush Toy to Direct Synthetic Characters. *Proc. CHI '99*.

16. Knep, B., Hayes, C., Sayre, R., and Williams, T. Dinosaur Input Device. *Proc. CHI 1995*, (1995), 304–309.

17. Mukai, T., Kuriyama, S. Geostatistical Motion Interpolation. *ACM Trans. Graph.* 24, 3 (2005), 1062-1070.

18. Raffle, H. S., Parkes, A. J., and Ishii, H. Topobo: A Constructive Assembly System with Kinetic Memory. *Proc. CHI '04*, (2004), 647–654.

19. Rose, C., Sloan, P., and Cohen, M. Artist-directed inverse-kinematics using radial basis function interpolation. *Computer Graphics Forum*, 20, (2001), 239–250.

20. Sekiguchi, D., Inami, M., and Tachi, S. RobotPHONE: RUI for Interpersonal Communication. *CHI '01 Extended Abstracts*, (2001), 277–278.

21. Shimizu, N., Koizumi, N., Sugimoto, M., Nii, H., Sekiguchi, D., and Inami, M. A Teddy-bear-based Robotic User Interface. *CIE* 4, 3 (July 2006), 8.

22. Tachi, S., Arai, H., and Maeda, T. Development of an Anthropomorphic Tele-existence Slave Robot. *Proc. ICAM '89*. JSME, (1989), 343–348.

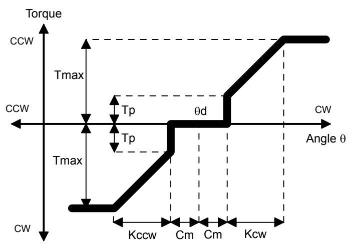
23. Weller, M. P., Do, E. Y., and Gross, M. D. Posey: Instrumenting a Poseable Hub and Strut Construction Toy. *Proc. TEI '08*, (2008), 39–46.

24. Wiley D.J., Hahn J.K. Interpolation Synthesis of Articulated Figure Motion. *IEEE CG&A*, 17, 6, (1997), 39-45.

APPENDIX: GRAVITY COMPENSATION DETAILS

The servomotors have an RS485 serial interface, and the host PC controls them by reading the current angle θ and setting the servo gain K_{cw}, K_{ccw} and target angle θ_d (cw means clock-wise and ccw means counter-clock-wise). The output torque T is produced by the servomotor according to the current angle θ, target angle θ_d, and the given parameters via Eq. (1).Other parameters are T_p: punch, T_{max}: maximum torque, C_m: compliance margin. See figure below.

$$\mathrm{T}^{t+1} = \begin{cases} \frac{T_{max} - T_p}{K_{cw}}(\theta^t - \theta_d^t - C_m) + T_p & (\theta_d^t - \theta^t > C_m) \\ 0 & (-C_m < \theta_d^t - \theta^t < C_m \\ \frac{T_{max} - T_p}{K_{ccw}}(\theta^t - \theta_d^t + C_m) + T_p & (\theta_d^t - \theta^t < -C_m \end{cases} \quad (1)$$



This will keep the difference in between the current angle θ and the target angle θ_d small unless a strong force is applied to the joint. If the difference exceeds a predefined threshold θ_{th}, the system assumes that the user intends to control the joint and updates the target angle θ_d as follows.

The torques T_{Tcw} and T_{Tccw} needed to overcome θ_{th} and thus start rotating the joint can be expressed as follows:

$$\theta_d^{t+1} = \begin{cases} \theta_d^t & (\theta_d^t - \theta^t > \theta_{th}) \text{ or } (\theta_d^t - \theta^t < -\theta_{th}) \\ \theta_d^{t+1} & (-\theta_{th} < \theta_d^t - \theta^t < \theta_{th}) \end{cases} \quad (2)$$

$$T_{Tcw}^t = \frac{T_{max} - T_p}{K_{cw}^t}(\theta_{th} - C_m) + T_p$$

$$T_{Tccw}^t = \frac{T_{max} - T_p}{K_{ccw}^t}(\theta_{th} - C_m) + T_p \quad (3)$$

T_{Tcw} and T_{Tccw} must be larger than gravitational forces T_{Gcw} and T_{Gccw} (one of which will be zero) applied to the joint in order to keep the current joint angle. The gravity forces are calculated via physical simulation. The force applied by the user's hand T_U is then obtained by subtracting T_G from T_T. We introduce a threshold T_{Uth} and start changing the θ_d when T_U exceeds T_{Uth}. So, we set T_{Tcw} and T_{Tccw} as follows

$$T_{Tcw}^t = T_{Gcw}^t + T_{Uth}$$

$$T_{Tccw}^t = T_{Gccw}^t + T_{Uth} \qquad (T_{Uth} \geq 0) \quad (4)$$

and then compute the gain as follows.

$$K_{cw}^{t+1} = \frac{T_{max} - T_p}{T_{Tcw}^t - T_p}(\theta_{th} - C_m)$$

$$K_{ccw}^{t+1} = \frac{T_{max} - T_p}{T_{Tccw}^t - T_p}(\theta_{th} - C_m) \quad (5)$$

This allows the system to compensate for gravity and yet be rotated via the small torque $T_U > T_{Uth}$ provided by the operator. As this gravity compensation technique is based on the difference between the current angle θ and the target angle θ_d, the cycle rate becomes critical if the user applies a quick rotation. Because our system is rather slow for haptic purposes, if the change in the target angle θ_F is greater than the threshold θ_{Fth}, the torque output is shut off to minimize the resistance.

In summary, the presudo code for the gravity compensation process is as followis

```
1. Read current angle θ
2. Calculate gravity force $T_{Gcw}$ and $T_{Gccw}$ by physical simulation
3. Calculate servo gain $K_{cp}$ and $K_{ccp}$
4. Store current $\theta_d$ ($\theta_d' = \theta_d$)
5. If $\theta_d - \theta > \theta_{th}$ then $\theta_d = \theta$
6. If $\theta_d' - \theta_d > \theta_{Fth}$ then torque off else torque on
7. Go to 1
```