Artif Intell Rev (2006) 25:37–53
DOI 10.1007/s10462-007-9042-5

Virtual human animation in natural language visualisation

Minhua Ma · Paul Mc Kevitt

Published online: 9 October 2007

Abstract Simulation motion of Virtual Reality (VR) objects and humans has experienced important developments in the last decade. However, realistic virtual human animation generation remains a major challenge, even if applications are numerous, from VR games to medical training. This paper proposes different methods for animating virtual humans, including blending simultaneous animations of various temporal relations with multiple animation channels, minimal visemes for lip synchronisation, and space sites of virtual human and 3D object models for object grasping and manipulation. We present our work in our natural language visualisation (animation) system, CONFUCIUS, and describe how the proposed approaches are employed in CONFUCIUS' animation engine.

Keywords 3D animation · Language visualisation · Temporal relations · Virtual human animation · Virtual Reality

1 Introduction

Simulating human motion and behaviour by computer is an active and challenging area. Existing virtual human animations are either controlled by precreated animations (Szarowicz and Francik 2004), e.g. hand-animated using authoring tools like 3D Studio Max, Maya, and Poser, or motion captured data, or dynamically generated by animation techniques such as inverse kinematics (IK). However, there is a lack of consideration for presenting temporal relations between multiple animation sequences and integrating different human animation

M. Ma (✉)
School of Computing & Information Engineering, Faculty of Engineering, University of Ulster,
Coleraine BT52 1SA, Northern Ireland
e-mail: m.ma@ulster.ac.uk

P. Mc Kevitt
School of Computing & Intelligent Systems, Faculty of Engineering, University of Ulster,
Derry/Londonderry BT48 7JL, Northern Ireland
e-mail: p.mckevitt@ulster.ac.uk

sequences to present simultaneous motions. Here, we propose an approach to present various temporal relations of virtual human actions (especially overlapped interval relations) using multiple animation channels. We also describe techniques used in lip synchronisation and virtual grasping. We present our work in the natural language visualisation (animation) system, CONFUCIUS, and show how these techniques are employed in CONFUCIUS' animation engine and achieve more flexibility and intelligence in generated animations.

First, in Sect. 2 we introduce the natural language visualisation system, CONFUCIUS and review various techniques for humanoid animation. Next in Sect. 3, the 13 interval temporal relations, in particular, overlapped relations which indicate simultaneous motions, are introduced, and the sense of iteration and its presentation are discussed. Then we propose the animation blending approach of using multiple animation channels, introduce three minimal visemes for lip synchronisation, and discuss using space sites of virtual human and 3D object models in virtual grasping and object manipulation in Sect. 4. Next, Sect. 5 compares our work with related work on humanoid animation blending, and finally, Sect. 6 concludes with a discussion of possible future work on integrating other animation generation techniques such as IK.

2 Background

We are developing a natural language visualisation system called CONFUCIUS, which automatically generates 3D animation and speech from natural language input as shown in Fig. 1. The dashed part in Fig. 1 is the knowledge base including language knowledge (lexicons and a syntax parser) which is used in the Natural Language Processing (NLP) module, and visual knowledge such as 3D models of characters, props, and animations of actions, which is used in the animation engine. The surface transformer takes natural language sentences as input and manipulates surface text. The NLP module uses language knowledge to parse sentences and analyse their semantics. The media allocator then generates an XML-based specification of the desired multimodal presentation and assigns content to three different media: animation, characters' speech, and narration, e.g. it sends the parts bracketed in quotation marks near a communication verb to the text-to-speech engine. The animation engine accepts semantic representations and uses visual knowledge to generate 3D animations. The outputs of the animation engine and the text-to-speech engine are combined in the synchronising module, which outputs a 3D virtual world including animation and speech in VRML. Finally, the narration integration module integrates the VRML file with the presentation agent, Merlin the Narrator, to complete a multimedia presentation. Currently, CONFUCIUS is able to visualise single sentences which contain action verbs with *visual valency* (Ma and Mc Kevitt 2004b) of up to three, e.g. "John left the gym", "Nancy gave John a loaf of bread".

2.1 Previous work on animation of virtual objects and humans

Animation of virtual objects and humans have experienced important developments in the last decade. Only a few existing approaches investigate the temporal relations between multiple animations and how to combine simultaneous animations (Campos et al. 2002; Perlin and Goldberg 1996). Campos et al. (2002) propose a set of operations to change the spatio-temporal configuration of animations of virtual objects in Geographic Information Systems (GIS) for geographic information visualisation and exploration, e.g. makeMeets, makeStarts, makeFinishes and makeEquals. The makeMeets operation of two actions produces a new animation where the second action starts immediately after the first. The makeStarts and makeFinishes operations of two actions produce a new animation where both actions start or finish at the same time. They are used to simulate a situation when the simultaneous occurrence of actions is desired while the original pace of the actions is preserved. The makeEquals operation of two actions produces a new animation when both actions start and finish at the same time. This operation changes the start point and duration of the second action, and the result is that the pace of the second action changes. However, this model doesn't take articulated virtual objects or virtual humans into account; the animation model only applies to geographic virtual objects such as buildings, cars and ships.

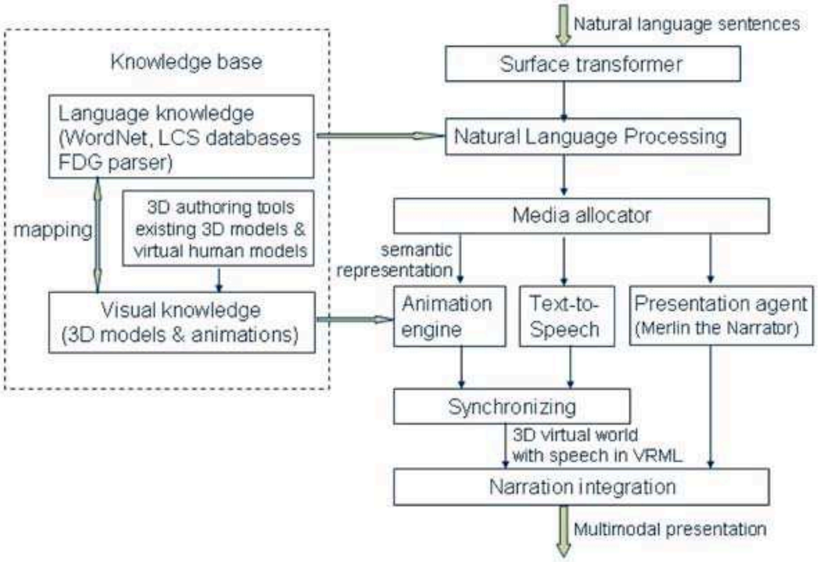

Natural language sentences
Knowledge base
Surface transformer
Language knowledge (WordNet, LCS databases FDG parser)
Natural Language Processing
mapping
3D authoring tools existing 3D models & virtual human models
Media allocator
semantic representation
Visual knowledge (3D models & animations)
Animation engine
Text-to-Speech
Presentation agent (Merlin the Narrator)
Synchronizing
3D virtual world with speech in VRML
Narration integration
Multimodal presentation

Fig. 1 Architecture of CONFUCIUS

Representing humanoid kinematics is another main component of VR animation. The kinematic animation techniques vary from simple application of precreated animation frame data (keyframes, either hand-animated or motion-captured), to complex on-the-fly inverse kinematics (IK). IK is a system in which the movement of the children is passed back up the chain to the parent in the hierarchical skeleton tree. Given a desired position and orientation for a final link in a hierarchy chain, IK establishes the transformations required for the rest of the chain. Animation is performed by affecting the ends of the chain, e.g. in biped walking animation, by moving the foot and the shin, knees and thighs rotate in response. A good overview of IK techniques can be found in Lander (1999). IK models the flexibility and possible rotations of joints and limbs in 3D creatures. IK adds flexibility which avoids canned motion sequences seen in keyframing animations, and hence enables having an infinitely expandable variety of possible animations available to a virtual character. The character control file is also reduced to a physical description of the character and a set of behavioural modifiers that are used to alter the flavour of the animations (Badler 1993).

In keyframing animation, animators have to explicitly define the key values of the character's joints at specific time instants, namely "key frames", to generating a motion. Then the key values are interpolated so that in-between frames are generated. CONFUCIUS' animation uses the keyframing technique for virtual human motions. The traditional approach of animating characters (agents/avatars) provides a set of animations from which the user/system

Table 1 Allen's 13 interval relations

Temporal relations	Example	Endpoints	Example sentences
1. Precede	xxxx	$x_e < y_s$	John left before Mary arrived.
2. Inverse precede	yyyy		
3. Meet	xxxx	$x_e = y_s$	All passengers died when the plane
4. Inverse meets	yyyy		crashed into the mountain.
5. Overlap	xxxxx	$x_s < y_s < x_e \cap$	Mary got up. She felt very ill.
6. Inverse overlap	yyyyy	$x_e < y_e$	
7. During	xxx	$x_s > y_s \cap$	John arrived in Boston last Thursday.
8. Include	yyyyyyyy	$x_e < y_e$	
9. Start	xxxx	$x_s = y_s \cap$	John has lived in Boston since 2000.
10. Inverse start	yyyyyyyy	$x_e < y_e$	
11. Finish	xxx	$x_e = y_e \cap$	John stayed in Boston till 2000.
12. Inverse finish	yyyyyy	$x_s > y_s$	
13. Equal	xxxxx	$x_s = y_s \cap$	John drove to London. During his
	yyyyy	$x_e = y_e$	drive he listened Classic FM.

"e" denotes "end point", "s" denotes "start point".

can select. In most current graphical chatrooms the user can control his avatar behavior by selecting an animation sequence from a list of available motions. The avatar can only play one animation at a time, i.e. only apply one precreated animation for the entire duration of the animation sequence.

IMPROV (Perlin and Goldberg 1996) uses procedural animation combined with behavioural scripting for creating flexible characters for virtual theatre. IMPROV divides the actions of avatars into a set of groups. The action, in this case, is defined as a single atomic or repetitive activity that does not require explicit higher-level awareness or conscious decisions. Actions within a group are mutually exclusive of one another; activating one causes the action currently active to end. Actions in different groups can operate simultaneously, so activities of certain parts of the body can be layered over those involving others. Using this structure, the basic set of actions can be combined to create dozens of composite animations while minimising the risk of inadvertently creating a behaviour that is either unbelievable or not lifelike. The solution serves as the mechanism for user-controlled avatars by enabling multiple levels of abstraction for the possible actions.

We have discussed state-of-the-art animation generation techniques especially virtual human animation and introduced CONFUCIUS' architecture. Our focus in this paper is on temporal relationships between human motions performed by one character and how to use multiple animation channels to present these temporal relationships.

3 Temporal relations between simultaneous animations

Table 1 lists Allen's 13 temporal relations (Allen 1983) that are used in visual semantic representation of verbs in CONFUCIUS' language visualisation (Ma and Mc Kevitt 2004a). Simultaneous animations playing on multiple channels of a virtual human are closely related to the overlapped temporal relations, i.e. the relations 5–13, in Table 1.

Iteration is a temporal factor affecting animated characters. Sense of iteration is not encoded in English syntax though it may be added by some prepositional phrases like "for

Table 2 Temporal boundedness of events

Examples	Temporal Boundedness	Prefixing “for hours” or “until midnight”
John slept.	Unbounded	Acceptable
John waked.	Bounded	Not acceptable
John entered the house.	Bounded	Not acceptable
John walked toward the house.	Unbounded	Acceptable
The light flashed.	Bounded (repeatable)	Acceptable, add the sense of repetition
Somebody hammered the door.	Bounded (repeatable)	Acceptable, enhance the sense of repetition

hours”, “until midnight”, or temporal quantifier such as *twice, three times, every*, and so forth. Consider, for example, the difference between the two sentences below.

```
John taught two hours every Monday. (iteration)
John taught two hours on Monday.
```

Jackendoff (1990) ascribes the sense of iteration to *temporal boundedness*. Table 2 shows some examples of temporal boundedness of events. Temporal bounded events (e.g. Table 2: 2, 3) are also called *punctual events* or *achievement* events (distinct from *accomplishment* events (Vendler 1967). The prepositional phrases “for hours” and “until midnight” can follow temporally unbounded processes, and place either a measure or a boundary on them. “John slept”, for instance, expresses an unbounded event, so it can be felicitously prefixed with these prepositional phrases. But “John waked” expresses a temporally bounded event, so it cannot be further measured or bounded by these prepositional phrases.

Some verbs have the sense of repetition included/hinted in their lexical semantics, e.g. Table 2: 5 and 6. Prefixing “for hours” or “until midnight” will add/enhance the sense of repetition to them. However, there is a nuance between 5 and 6. Without those prepositional phrases, “the light flashed” means it flashed once, whereas “Somebody hammered the door” suggests (s)he hammered the door repeatedly. Therefore, “for hours” *adds* the sense of repetition in 5, and *enhances* it in 6. Example 2 and 3 are bounded but unrepeatable, so they cannot give grammatical productions when prefixing “for hours” or “until midnight”.

Jackendoff (1990) thinks that the operator, which maps a conceptual constituent that encodes a single event into a conceptual constituent that encodes a repeated sequence of individual events of the same type, has the same semantic value as the plural marker, which maps a conceptual constituent that encodes an individual thing into a conceptual constituent that encodes a collection of things of the same type, to wit, the bounded/unbounded distinction in events is strongly parallel to the count/mass distinction in noun phrases (NPs). The criterion for the boundedness and countableness distinction has to do with the description of parts of an entity. For instance, a part of “an apple” (count) cannot itself be described as “an apple”, but any part of a body of “water” (mass) can itself be described as “water”; a part of the event “John entered the house” (bounded) cannot itself be described as “John entered the house”, but any part of “John walked toward the house” (unbounded) can be described as “John walked toward the house”. Therefore, a static graphic scene can only represent unbounded events such as “John walked toward the house” properly, by selecting a representative part of the event; while bounded events are better presented by animation.

Distinction for sense of iteration is very important for visualising events in CONFUCIUS since the animation generator needs to know whether it’s necessary to repeat an action loop, and whether it’s necessary to animate the complete process of an event (a bounded event) or just a part of it (an unbounded event).

Table 3 Verbs defined by repeatable subactivities

Repetition of subactivities	Repetition of subactivities	Number of iterations
walk():- $[\text{step}()]_{\text{R}}$.	hammer(. . .):- $[\text{hit}(\ldots,\text{hammer})]_{\text{R}}$.	recalculate():- $[\text{calculate}()]_{2}$.

CONFUCIUS' semantic representation has a facility to represent repeatable periods of subactivities. Square brackets and a subscript R are used to indicate the repetition constructs in the examples given in Table 3, which can also be nicely captured by Kleene iteration in finite state descriptions for temporal semantics. The activities bracketed by $[\,]_R$ are repeatable. Besides periodical repetition of subactivities, it can represent morphological prefix "re-" as well, as the "recalculate" example in Table 3, substituting the number of iterations (which is 2 in this case) for R. This facility of representing iteration may be used for post-lexical level repetition, such as events marked by "again", "continues to", or "a second time".

Animation loops are used to present action repetition. This facility indicates whether the played animation should loop. If not specified, the animation will loop, i.e. looping is enabled as default, which is controlled by a time sensor in the VRML file.

4 Animating virtual humans

CONFUCIUS uses the H-Anim standard (H-Anim 2001) for character modelling and animation. H-Anim is a VRML97 representation for humanoids. It defines standard human *Joints* articulation (e.g. knee and ankle), *Segments* dimensions (e.g. thigh, calf and foot), and *Sites* (e.g. hand_tip, foot_tip) for "end effector" and attachment points for clothing. An H-Anim file contains a joint-segment hierarchy as shown in Fig. 2. Each joint node may contain other joint nodes and a segment node that describes the body part associated with the joint. Each segment is a normal VRML transform node describing the body part's geometry and texture. H-Anim humanoids can be animated using keyframing, IK, and other animation techniques.

Since our task of language animation in CONFUCIUS focuses on off-line generation, and real-time interaction is never our concern, we adopt the H-Anim standard to model the virtual characters in our language visualisation. H-Anim provides four Levels of Articulation (LOA) for applications which require different *levels of detail*. Some applications such as medical simulation and design evaluation require high fidelity to anthropogeometry and human capabilities, whereas computer games, training and visualised living communities are more concerned with real-time performance. Natural language visualisation is not usually concerned with accurate simulation of humans. We use Level 2 of Articulation (LOA2) of H-Anim in character modelling for CONFUCIUS. This level ensures enough joints for human movements in language visualisation, e.g. it includes enough hand joints for grasp postures. Figure 2 illustrates the joints of LOA2.

Based on Babski's (2000) animation prototype, we design our virtual characters' animation which is capable of being applied to various human models with different LOAs. Needed ROUTEs are generated dynamically based on the joint list of the H-Anim body and the joint list of the animation. Figure 3 shows an example external prototype inserted at the end of a virtual character's H-Anim file by the animation engine. It uses keyframe information in the external VRML file "..\animation\walk.wrl" to make the virtual character walk.

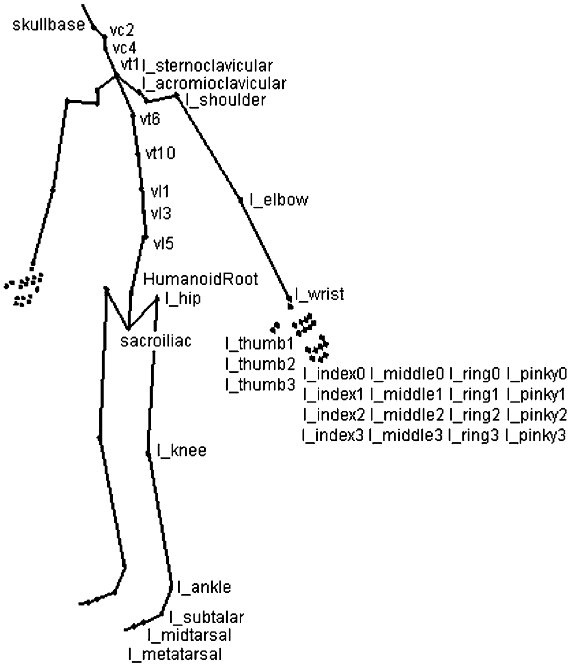


Fig. 2 H-Anim joint hierarchy

```
# -------- inserted by CONFUCIOUS animation engine ---
EXTERNPROTO Behaviour [
      eventIn SFTime LaunchAnim
      exposedField SFTime set_startTime
      exposedField SFTime set_stopTime
      field MFNode HumansList
]"..\animation\walk.wrl"
DEF behv Behaviour {
      HumansList [
            USE humanoid
      ]
}
ROUTE hanim_BodyTouch.touchTime TO behv.LaunchAnim
```

Fig. 3 External prototype of H-Anim animation

The animation file defines keyframes of all `OrientationInterpolator` and `PositionInterpolator` involved in the movement. The Script node dynamically adds ROUTEs according to the list specified in InvolvedJointNameList and `InvolvedJointPtrList` in the animation file. The matching between the animation and the body is performed by using the joints list in the humanoid prototype. Therefore, `InvolvedJointNameList` must have a one-to-one matching to the humanoid `joints` list defined in the virtual character's geometry file. If the animation is applied to a lower LOA character, e.g. LOA1, and a joint is not implemented, the corresponding field should be a dummy Transform/Joint node.

4.1 Simultaneous animations and multiple animation channels

Performing simultaneous animations is not a problem for the lower level procedural human animation modeling languages, e.g. VHML (Marriott et al. 2001), an XML-based human

```
<left-calf-flex amount=”medium”>
<right-calf-flex amount=”medium”>
  <left-arm-front amount=‟medium">
  <right-arm-front amount=‟medium">
    Standing on my knees I beg you pardon
  </right-arm-front>
  </left-arm-front>
</right-calf-flex>
</left-calf-flex>
```

a A VHML example

```
script(walk_forward_step(Agent),ActionList):-
  ActionList=[parallel(
                  [script_action(walk_pose(Agent)),
                   move(Agent,front,fast)]
                     )].
```

b A STEP example

Fig. 4 Representing parallel temporal relation

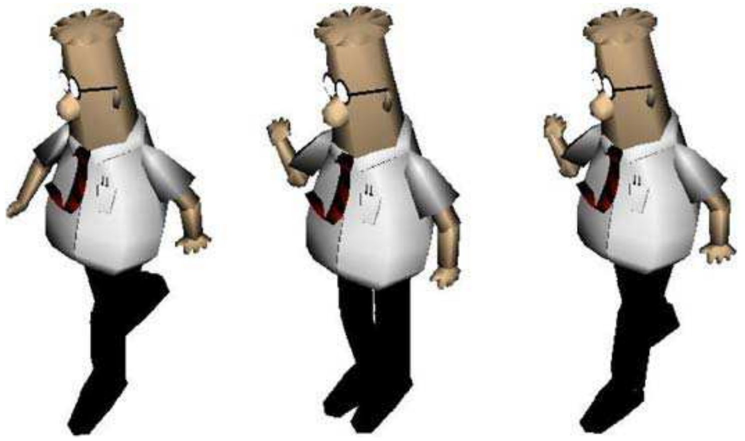
Table 4 The animation registration table

Animations	Sacroiliac	l_hip	r_hip	...	r_shoulder
Walk	2	2	2	...	1
Jump	2	2	2	...	1
Wave	0	0	0	...	2
Run	2	2	2	...	1
Scratch head	0	0	0	...	2
Sit	2	2	2	...	1
...	...	...	...	...	...

animation language, and STEP (Huang et al. 2002), a Prolog-like human animation script language, since they provide a facility to specify both sequential and parallel temporal relations. Figure 4 shows how VHML and STEP represent the parallel temporal relation. However, simultaneous animations cause the Dining Philosopher's Problem (Dijkstra 1971) for higher level animation using pre-defined animation data, i.e. multiple animations may request to access same body parts at the same time. In order to solve this problem, we introduce the approach of multiple animation channels to control simultaneous animations.

A character that plays only one animation at a time has only a single channel, while a character with upper and lower body channels will have two animations playing at the same time. Multiple animation channels allow characters to run multiple animations at the same time such as walking with the lower body while waving with the upper body. Multiple animation channels often need to disable one channel when a specific animation is playing on another channel to avoid conflicts with another animation.

We use an animation table as shown abridged in Table 4 to implement multiple animation channels. Every pre-defined animation must register in the animation table and specify which joints are used for the animation. In Table 4, each row represents one animation, and each column represents one joint involved. 0 indicates that the joint is not used for the animation; 1 indicates that it is used and can be disable when playing simultaneous animations; and 2 means that the joint is used and cannot be disabled. When simultaneous animations are requested, the animation engine checks the animation table and finds if the involved joints



a Walking **b** Waving **c** Walking and waving

Fig. 5 An example of motion integration **(a)** Walking **(b)** Waving **(c)** Walking and Waving

of these animations conflict, i.e. if there is any joint whose values for both animations are 2, these animations conflict and they cannot be played at the same time. If two animations do not conflict (for example, “run” and “throw”), the animation engine merges their keyframes information, i.e. interpolators, and creates a new animation file which will be applied to the virtual human.

Figure 5 shows an example of integrating the two animations “walk” and “wave”. The first figure is a snapshot of walking animation, the second is waving animation, and the third animation is integrated from walking and waving, using the multiple animation channels approach. The motion of waving only uses three rotation interpolators: r_shoulder, r_elbow, r_wrist. The animation engine looks up the animation table and finds that the walking animation also uses these three joints and their values are all 1, which means the right arm movements of walking can be disabled and overwritten by the movements of waving. The animation engine then replaces the keyframes of these three joints in the walking animation file with those in the waving file and generates an integrated motion. This approach allows us to take advantage of procedural animation effects in the same manner as regular animations, adding an additional level of flexibility and control when animating virtual characters.

4.2 Facial expression and lip synchronisation

Lip movement concerns another modality (speech) by creating the illusion of corresponding speech. Traditional animators use a system called track reading in which the animation is carefully analysed for mouth positions laid out against a time sheet. The animator’s true skill is knowing how to condense speech into as few positions of lips as needed to create the speaking illusion.

Using lip movement to support speech output helps to alleviate communication problems by redundant coding. Previous user interface agents focus on the visualisation of a fully animated talking head (Alexa et al. 2000; CSLU 2006). Facial expression can be implemented by using `CoordinateInterpolator and NormalInterpolator in` VRML to animate morphing and shading on a character’s face. MPEG4 defines 14 visemes and six expressions represented by low level facial animation parameters (FAPs), which are

Table 5 MPEG4 visemes

Viseme	Displacer Name	Phonemes	Example
1	Viseme_pbm	p, b, m	put, bed, mill
2	Viseme_fv	f, v	far, voice
3	Viseme_th	T, D	think, that
4	Viseme_td	t, d	tip, doll
5	Viseme_kg	k, g	call, gas
6	Viseme_ts	tS, dZ, S	chair, join, she
7	Viseme_sz	s, z	sir, zeal
8	Viseme_nl	n, l	lot, not
9	Viseme_r	r	red
10	Viseme_a	A:	car
11	Viseme_e	e	bed
12	Viseme_i	I	tip
13	Viseme_q	Q	top
14	Viseme_u	U	book

Table 6 MPEG4 expressions

#	Expressions	Textual description
1	Joy	The eyebrows are relaxed. The mouth is open and the mouth corners pulled back toward the ears.
2	Sadness	The inner eyebrows are bent upward. The eyes are slightly closed. The mouth is relaxed.
3	Anger	The inner eyebrows are pulled downward and together. The eyes are wide open. The lips are pressed against each other or opened to expose the teeth.
4	Fear	The eyebrows are raised and pulled together. The inner eyebrows are bent upward. The eyes are tense and alert.
5	Disgust	The eyebrows and eyelids are relaxed. The upper lip is raised and curled, often asymmetrically.
6	Surprise	The eyebrows are raised. The upper eyelids are wide open, the lower relaxed. The jaw is opened.

represented as a set of feature points on the face. Tables 5 and 6 list H-Anim suggested displacer nodes which are taken from MPEG4 FAPs.

Each FAP is controlled by a specific muscle (e.g. eyes, lips, jaw, brows). Visemes usually concern lips and jaw movement, and expressions concern lips, eyes and eyebrows. We distinguish three visemes (Table 7) and the six expressions of MPEG4 definition using parameters of eyes, eyelids, brows, lips, and jaw. We ignore all consonant visemes because they are not distinct enough for a rough simulation and have high computational costs. The five vowel visemes defined in MPEG4 (Table 5, 10–14) are merged to three visemes according to the two articulation features which can be shown via jaw and lip movement: (1) high/low tongue position (jaw close/open), and (2) lip roundness. The three visemes are shown as the three bounded areas in the cardinal vowel quadrilateral in Fig. 6. Viseme *a* is an open-jaw

Table 7 CONFUCIUS' visemes

CONFUCIUS visemes	MPEG4 Visemes	Examples
Viseme a	Viseme_a Viseme_e	Car Bed
Viseme i	Viseme_i	Tip, tea
Viseme o	Viseme_q Viseme_u	Top Book

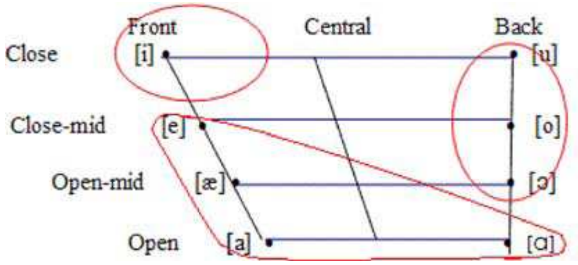
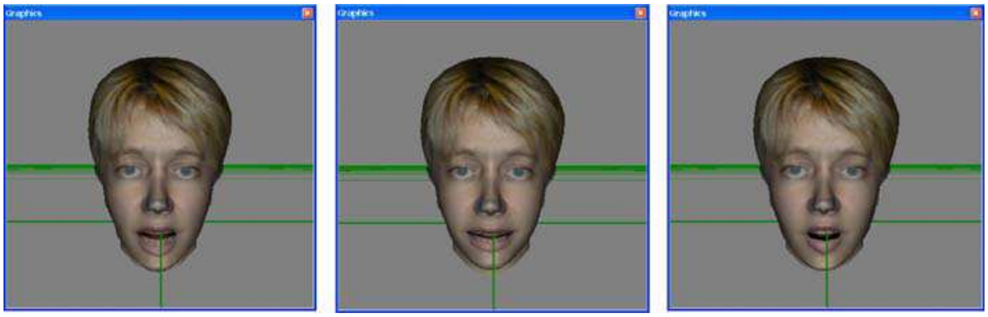


Fig. 6 Cardinal vowels quadrilateral



a Viseme a **b** Viseme i **c** Viseme o

Fig. 7 Lip synchronisation of CONFUCIUS' viseme a, i, o

articulation; viseme *i* is a close-jaw, extended-lip articulation; and viseme *o* is a rounded-lip articulation. Figure 7 shows lip synchronisation of these three visemes.

It is computationally economical to introduce Level-Of-Detail (LOD) for facial expression and lip synchronisation to accelerate the generation and rendering of virtual humans. Simplifications of the virtual human, i.e. omitting facial animation, are produced and contain fewer details. These simplifications can then be used when the virtual human is further away and the facial details are not noticed anyway.

4.3 Space sites of virtual humans

In the geometric VRML files of 3D objects and H-Anim files of virtual humans, there are lists of grasp sites and their purposes, and intrinsic directions such as top and front, defined with respect to an object, and sites for manipulating and placing/attaching objects defined with respect to a virtual human. We classify three types of objects as follows:

1. Small props which are usually manipulated by hands or feet, e.g. cup, box, hat, ball.
2. Big props which are usually sources or targets (goals) of actions, e.g. table, chair, tree.

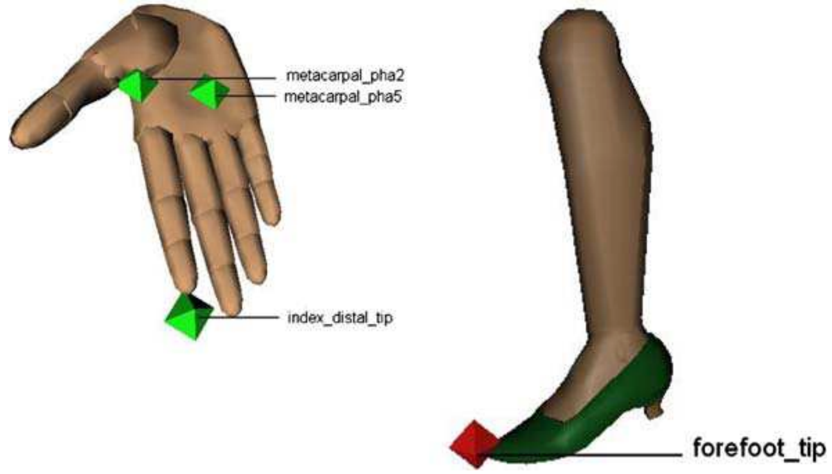


Fig. 8 Site nodes on the hands and feet of a virtual human

3. Stage props which have internal structure, e.g. house, restaurant, chapel.

To figure out where to place these three types of props around virtual human bodies, we create corresponding site tags for virtual humans using H-Anim Site nodes.

(1) *Manipulating small props* For manipulation of small props, a virtual human has six sites on the hands (three sites for each hand, l_metacarpal_pha2, l_metacarpal_pha5, l_index_distal_tip, r_metacarpal_pha2, r_metacarpal_pha5, r_index_distal_tip), one site on the head (hanim_skull_tip), and one site for each foot tip (l_forefoot_tip, r_forefoot_tip). The sites metacarpal_pha2 are used for grip and pincer grip; metacarpal_pha5 are for pushing; and index_distal_tip are for pointing. The sites forefoot_tip are for kicking. Figure 8 shows the position of these sites.

(2) *Placing big props* For big props placement, we use five sites indicating five directions around the human body: x_front, x_back, x_left, x_right, x_bottom. We leave out x_top because there is already a site node, hanim_skull_tip, defined on the head of every virtual human for attaching headdress. Big props like a table or chairs are usually placed at these positions.

(3) *Setting stage props* For stage props setting, we have five more space tags besides those in (2) around a virtual human to indicate further places: x_far_front, x_far_back, x_far_left, x_far_right, x_far_top. Figure 9 shows the positions of these sites. Stage props such as a house often locate at these far sites of virtual humans.

4.4 Space sites of 3D objects and grasping hand postures

The geometric file of a 3D object define its shape, default size, functions, as well as any constraints that might be associated with the manipulation of that particular object, such as allowable actions that can be performed on it, and what the expected outcome of the actions will be, e.g. the outcome state of a lamp when it is switched on/off.

Similar to virtual humans, objects in the graphic library usually have six space sites indicating six directions around the object if applicable: x_front, x_back, x_left, x_right, x_on, x_under, one functional space site x_in, and several grasp site-purpose pairs. Stage props, such as “house”, “restaurant”, “chapel”, normally don’t have grasp site-purpose pairs.

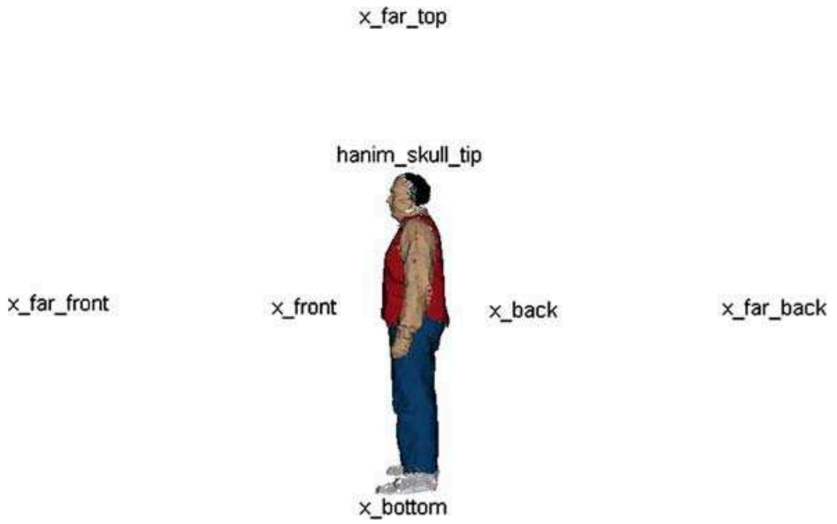


Fig. 9 Site nodes around a virtual human's body

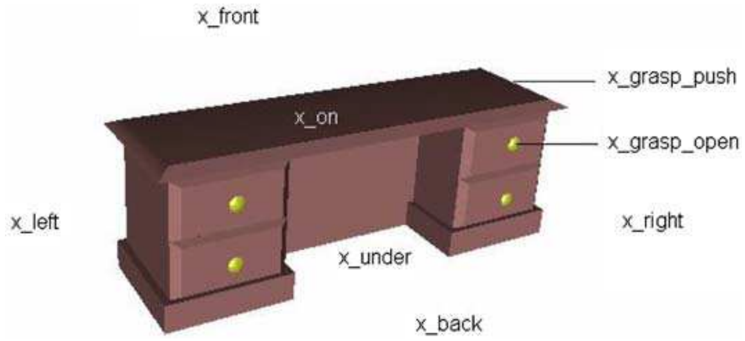


Fig. 10 Space sites of a 3D desk

Figure 10 illustrates sites of a desk which has two grasp site-purpose pairs. One on the drawer knob for opening, and the other at the side of the desk for pushing. Space sites often relate to the object's function, for instance, the front and back sites of a desk or a chair depend on their functionalities. Objects' space sites are not only useful for objects/virtual human positioning, but also for expected virtual human behaviours in order to accomplish the interaction with them. For example, before opening a drawer of the desk in Table 9, the actor is expected to be in a suitable position (i.e. x_back) so that the drawer will be in the reach and not collide with the virtual human when opening.

Table 9 gives a list of verbs describing hand movements. Some of them (e.g. verbs of empty-handed gestures and haptic exploration) can be defined in the animation library, while others cannot be defined solely on the verbs because their hand shapes are closely associated with the shape, size, or functionality of the object they manipulate. Cadoz (1994) defined the later group as *ergotic* hand movements, which are associated with the notion of work and the capacity of humans to manipulate the physical world and create artefacts. For example, the hand shape and movement of "wave" is defined in an animation key frame file wave.wrl, but the hand shape of "pick up" is uncertain if we don't know what the virtual human picks up because the hand shapes and grasp points of picking up a cup and a bottle are quite different.

Table 8 Taxonomy of ergotic hand movements

Classification standards	Physical characteristics	Functions
Classes	• Change effectuated: position, orientation,shape • How many hands are involved: one or two • Indirection level: direct manipulation or through another object or tool	• Prehensile • Non-prehensile

Table 9 Verbs of hand movements

Ergotic hand movements

- Contact with the object: grasp, seize, grab, catch, embrace, grip, lay hold of, hold, snatch, clutch, take, hug, cuddle, cling, support, uphold
- Contact and changing position: lift, move, heave, raise, translate, push, pull, draw, tug, haul, jerk, toss, throw, cast, fling, hurl, pitch, depress, jam, thrust, shake, shove, shift, shuffle, jumble, crank, drag, drop, pick up, slip, hand over, give
- Contact and changing orientation: turn, spin, rotate, revolve, twist
- Contact and changing shape: mold, squeeze, pinch, wrench, wring, stretch, extend, twitch, smash, thrash, break, crack, bend, bow, curve, deflect, tweak, spread, stab, crumble, rumple, crumple up, smooth, fold, wrinkle, wave, fracture, rupture
- Joining objects: tie, pinion, nail, sew, button up, shackle, buckle, hook, rivet, fasten, chain up, bind, attach, stick, fit, tighten, pin, wrap, envelop, swathe
- Indirect manipulation (via other objects): cut, whet, set, strop, whip

Empty-handed gestures

- wave, snap, point, urge, show, size, count

Haptic exploration

- beat, bump, brush, caress, clink, drub, flick, fondle, hit, jog, kick, knock, nudge, pluck, prick, poke, pat, rap, rub, slam, slap, strike, stroke, struck, strum, tap, touch, thrum, twang, twiddle, throb, thwack, tickle, wallop, whop

Ergotic hand movements can be classified according to physical characteristics or their function (Table 8). Ergotic verbs in Table 9 are grouped by physical characteristic, i.e. change effectuated and indirection level. It is more common to classify ergotic hand movements according to their function, either prehensile or non-prehensile. Non-prehensile movements include pushing, lifting, tapping and punching. To illustrate how complex it can be to perform a simple task of ergotic hand movement, let's consider the example of picking up a mug: walking to approach the mug, deciding which hand to use, searching for the graspable site (i.e. the handle), moving body limbs to reach the handle, deciding which hand posture to use, adjusting hand orientation and the approaching aperture, grasping, close the grip, and finally lifting the mug.

There are two approaches to organising the knowledge required in the above task to achieve "intelligence" for successful grasping. One is to store applicable objects in the animation file of an action and using lexical knowledge of nouns to infer hypernymy relations between objects. For instance, one animation file of "pick-up" specifies the applicable objects are cups. The hand posture and movement of picking up a cup are stored in the animation file. From the lexical knowledge of the noun "mug" the system knows that a "mug" is a kind of "cup" and its meronymy relations (i.e. the "parts of" relationship. The meronyms of "mug", for example, are handle, stem, brim, and base), and the system then accesses the mug's geometric file to find its grasp site, i.e. the location of the handle. The system then combines the "pick up" animation for a cup object with the virtual human and uses it on the mug.

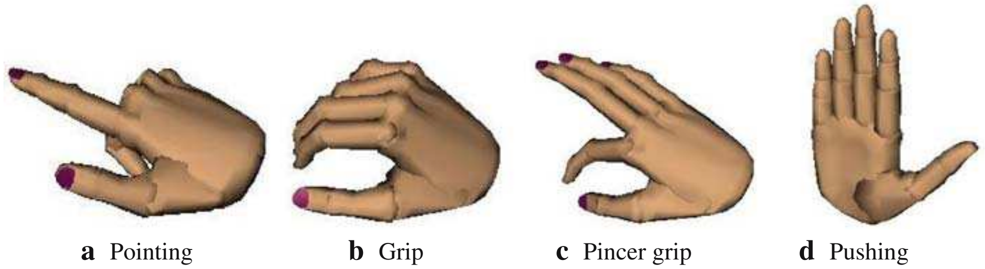


Fig. 11 Four hand postures for physical manipulation of objects **(a)** Pointing **(b)** Grip **(c)** Pincer grip **(d)** Pushing

The other approach includes the manipulation hand postures and movements within the object description, besides its intrinsic object properties. Kallmann and Thalmann (2002) call these objects "smart objects" because they have the ability to describe in details their functionality and their possible interactions with virtual humans, and are able to give all the expected low-level manipulation actions. This approach decentralises the animation control since object interaction information is stored in the objects, and hence most object-specific computation is released from the main animation control. The idea comes from the object-oriented programming paradigm, in the sense that each object encapsulates data and provides methods for data access.

Robotics techniques can be employed for virtual hand simulation of ergotic hand movements, as for automatic grasping of geometrical primitives. They suggest three parameters to describe hand movements for grasping: hand position, orientation and grip aperture. Su and Furuta (1994) suggest touching, pointing and gripping as a minimal set of gestures that need to be distinguished.

We use four stored hand postures and movement (Fig. 11) for moving, touching and interacting with 3D objects: index pointing (Fig. 11a, e.g. press a button), grip (Fig. 11b, e.g. hold cup handle, knob, or a cylinder type object), pincer grip (Fig. 11c, i.e. use thumb and index finger to pick up small objects), and palm push (Fig. 11d, e.g. push big things like a piece of furniture). They use different hand sites to attach objects. Hand postures and movements are defined as the motions of fingers and hands in virtual humans' VRML files. Different kinematic properties, such as movement velocity and grip aperture are fixed since further precision might involve significant costs in terms of processing time and system complexity but the result is only a little more realistic.

5 Relation to other work

Different approaches to blending multiple animations and representing various temporal relationships between human motions, many of which have limitations of requiring artists' involvement, are used in character animation. By narrowing the field further down to solving the conflicts between simultaneous animations, there is little directly related work. Compared with Perlin and Goldberg's (1996) grouping method, CONFUCIUS' multiple animation channels approach for integrating simultaneous motions provides a finer integration of simultaneous animations, and hence achieves more flexibility and control on virtual character animation. CONFUCIUS tackles limitations of the previous work and provides an integrated framework for effective integration of multiple simultaneous human motions.

In terms of lip synchronisation, CONFUCIUS' face animation introduces three visemes according to the two articulation features of jaw and lip movement and condenses speech into fewer lip positions without a detriment to the speaking illusion, compared with previous talking heads (Alexa et al. 2000; CSLU 2006) and standards like MPEG4 FAPs.

6 Conclusion and future work

Virtual human animation has a variety of applications in domains such as medical, training, interface agents and virtual reality games. The animations in these systems are either controlled by precreated animations (Szarowicz and Francik 2004), e.g. hand-animated keyframes or motion captured data, or dynamically generated by animation techniques such as inverse kinematics. However, few of these systems take into consideration temporal relations between multiple animation sequences, and integrate different human animation sequences to present simultaneous motions.

In CONFUCIUS, we use general-purpose virtual humans and their behaviours which follow the industry standard (H-Anim) and balance tradeoffs between computational efficiency and accuracy to produce believable human motions. We have investigated various temporal relations between human motions performed by one animated character, and employed multiple animation channels to integrate non-conflict simultaneous motions. This approach combines precreated and dynamically generated (procedural) animation facilities into a unified mechanism, and focusses on blending simultaneous animations. To simulate virtual object manipulation such as grasping, we use object-oriented object models to encapsulate object-related information, such as geometry, behaviour, space sites, and human-object interaction, to decentralise the control of animation engine. In addition, we propose three minimal vowel visemes for low-cost lip synchronisation. We believe these technique have the potential to have an impact not only on natural language visualisation but also on various areas such as computer games, movie/animation production, and intelligent agents.

Future research may address changing the original pace of one animation for blending simultaneous actions which has been mentioned in the makeEquals operation of virtual object animation (Campos 2002), e.g. to animate `jump ≡ kick`, one original pace of the animations has to be changed to suit the duration of the other animation if the original durations of the two animations are not equal. In addition, a long-term goal is to integrate a physics engine in order to simulate more realistic virtual grasping and virtual object behaviours.

Acknowledgements We are grateful for useful ideas and constructive comments on an earlier version of this paper from Dr. Michael McNeill of the School of Computing and Information Engineering, University of Ulster and Prof. Fionn Murtagh of the Department of Computer Science, Royal Holloway, University of London. We also would like to thank the two anonymous reviewers for their valuable suggestions.

References

Alexa M, Behr J, Miller W (2000) The morph node. In: Proceedings of Web3d/VRML 2000. Monterey, pp. 29–34
Allen JF (1983) Maintaining knowledge about temporal intervals. Commun ACM 26(11):832–843
Babski C (2000) Humanoids on the Web. Ph.D. Thesis, Computer Graphics Lab (LIG), Swiss Federal Institute of Technology (EPFL)
Badler NI, Philips CB, Webber BL (1993) Simulating humans. Oxford University Press
Cadoz C (1994) Les realites virtuelles. Dominos-Flammarion, Paris

Campos J, Hornsby K, Egenhofer M (2002) A temporal model of virtual reality objects and their semantics. In: DMSA 2002: eighth international conference on distributed multimedia systems. Redwood City, September, pp. 581–588

CSLU (2006) http://cslu.cse.ogi.edu/toolkit/index.html

Dijkstra EW (1971) Hierarchical ordering of sequential processes. Acta Informatica 1(2):115–138

H-Anim (2001) Humanoid animation working group. http://www.h-anim.org

Huang Z, Eliens A, Visser C (2002) STEP: a scripting lan-guage for embodied agents. In: Proceedings of the workshop on lifelike animated agents, Tokyo, pp. 46–51

Jackendoff R (1990) Semantic structures. Current studies in linguistics series. MIT Press, Cambridge

Kallmann M, Thalmann D (2002) Modeling behaviors of interactive objects for real time virtual environments. J Visual Lang Comput 13(2):177–195

Lander J (1999) Inverse kinematics for real-time games. In: Game developer conference 1999. http://www.darwin3d. com/confpage.htm

Ma M, Mc Kevitt P (2004a) Interval relations in visual semantics of verbs. Artif Intell Rev 21(3–4):293–316, Special Issue on Research in Artificial Intelligence and Cognitive Science (AICS-04), Kluwer-Academic Publishers, Dordrecht

Ma M, Mc Kevitt P (2004b) Visual semantics and ontology of eventive verbs. Natural language processing—IJCNLP-04. In: Su K-Y, Tsujii J-I, Lee J-H, Kwong OY (eds) First international joint conference, Hainan Island, China, Lecture Notes in Artificial Intelligence March 22–24, pp. 187–196, (LNAI) series, LNCS 3248. Springer Verlag, Berlin

Marriott A, Beard S, Stallo J, Huynh Q (2001) VHML: virtual human modelling language. http://www.vhml.org

Perlin K, Goldberg A (1996) Improv: a system for scripting interactive actors in virtual worlds. In: Proceedings of the 23rd annual conference on computer graphics and interactive techniques SIGGRAPH '96. ACM Press, New York, pp. 205–216

Su A, Furuta R (1994) A specification of 3D manipulations in virtual environments. In: Proceedings of the fourth international symposium on measurement and control in robotics: topical workshop on virtual reality. NASA Conference Publication 10163, Houston, November, pp. 64–68

Szarowicz A, Francik J (2004) Human motion for virtual people, In: International conference on computer games: artificial intelligence, design and education. CGAIDE, Reading

Vendler Z (1967) Linguistics and philosophy. Cornell University Press, Ithaca