

Neo Euclide: un sistema per l'animazione digitale real-time

Giovanni Civati, Samuele Vacchi, Daniele Marini*, Alessandro Rizzi

Dept. of Information Technology - University of Milano/Italia

E-mail: *gcivati@crema.unimi.it*

E-mail: *svacchi@crema.unimi.it*

E-mail: *rizzi@dti.unimi.it*

***Dept. of Information Science - University of Milano/Italia**

E-mail: *marini@dsi.unimi.it*

Abstract

In questo articolo presentiamo un sistema flessibile per la Performance Animation, generico, facile da usare e a basso costo.

Il sistema non è stato creato per uno scenario specifico, ma per animare creature e ambienti virtuali generici. Abbiamo deciso di supportare anche tecnologia a basso costo basando il software su librerie grafiche comuni e la valutazione delle performance su acceleratori grafici normalmente installati su PC e dispositivi di motion-capture acquistabili anche con budget limitato.

Introduzione

La Performance Animation è un ramo dell'informatica basato sullo studio dell'animazione real-time di modelli virtuali attraverso *motion-capture*.

Durante la Performance Animation i seguenti passi devono essere eseguiti per ogni frame:

- campionamento dei dati provenienti dai sensori di movimento (*motion-capture*)
- mappatura dei dati alla scena 3D
- rendering della scena 3D

La qualità dell'animazione dipende dal corretto allineamento tra i dati campionati nel primo passo ed ogni singolo elemento all'interno della scena 3D.

La maggior parte degli animatori utilizza espressioni procedurali e animazioni predefinite per ridurre la complessità del proprio lavoro. Quest'operazione copre un intero campo di ricerca [11,12,13]. Nella Performance Animation più è elevato il numero di movimenti messi a disposizione dal modello virtuale, più esso appare vivo, e conseguentemente l'animatore potrà renderlo più espressivo [1,2].

E' possibile rendere più autonomo il modello virtuale inserendo simulatori fisici, algoritmi di intelligenza artificiale e attraverso relazioni geometriche. Nell'intero processo, è il rendering della scena la parte che necessita più potenza di calcolo, dipendentemente dalla complessità stessa della scena.

Durante la creazione di modelli virtuali per la Performance Animation è necessario raggiungere compromessi tra qualità dell'immagine, espressività e facilità di utilizzo degli stessi.

La Performance Animation differisce sostanzialmente dalle tecniche tradizionali di animazione, come la *keyframe-animation* in cui l'animatore deve muovere ogni singolo elemento del modello virtuale, uno per volta, per raggiungere la posizione desiderata.

Con questa tecnica è possibile ottenere un risultato molto preciso, ma bisogna spendere molto tempo. Diversamente la Performance Animation permette di realizzare le animazioni molto velocemente, muovendo contemporaneamente ogni parte del personaggio virtuale [3]. Il feedback immediato di questa tecnica da la possibilità all'animatore di sincronizzare ogni movimento con gli elementi della scena 3D.

Nella creazione di una sequenza la Performance Animation può essere usata come base per l'animazione e successivamente si possono utilizzare tecniche più precise per raffinare ogni singolo movimento. Inoltre il motion-

capture sostituisce le curve matematiche di interpolazione con i campionamenti dei movimenti reali [4].

In una prestazione dal vivo il modello virtuale, guidato dall'animatore, può interagire direttamente con il pubblico, come se fosse vivo; un modello 3D può essere reso interattivo con l'ausilio di sensori ottici, creando un reale ambiente interattivo che permette all'animatore di reagire con naturalezza agli eventi.

Durante l'animazione dal vivo, dove abbiamo sperimentato l'uso del nostro sistema chiamato "Neo Euclide", l'animatore può vedere il pubblico attraverso una videocamera ed il personaggio virtuale può parlare ed interagire con le persone. L'idea di un personaggio virtuale che parla con naturalezza a persone reali è molto utile per studiare le reazioni delle persone in presenza di un'interfaccia virtuale di questo tipo.

Il sistema proposto: "Neo Euclide"

La Performance Animation è sempre stata basata su sistemi chiusi e dedicati, sfruttando dispositivi molto costosi con sensori ad alta precisione e super-calcolatori dedicati al rendering. Con l'avvento dei nuovi acceleratori grafici è possibile ottenere ottimi risultati dando supporto alla tecnologia di basso costo.

"Neo Euclide" è un sistema generico per la Performance Animation che integra un motore di real-time rendering proprietario (basato su tecnologia DirectX) e gestisce dispositivi di input interpretandone i segnali. Il sistema può essere inoltre interfacciato ad altri sistemi per l'acquisizione di segnali di input.

"Neo Euclide" è stato pensato per l'animazione dal vivo, non solo di creature ma di qualsiasi scena 3D creata nell'ambiente di modellazione. Sono supportati dispositivi standard (mouse, tastiere e joystick), guanti virtuali (General Reality 5th Glove, Immersion Cyberglove, Humanware Humanglove), strumenti MIDI e generatori software di dati. Il protocollo MIDI è stato completamente gestito, l'animatore può usare quindi qualsiasi dispositivo o applicativo in grado di produrre messaggi compatibili, come tastiere musicali o file MIDI. In questo modo abbiamo reso possibile l'interazione con altri sistemi e la registrazione dei dati.

Relazione con gli strumenti di modellazione

I modelli virtuali, che il sistema dovrà gestire ed animare, vengono realizzati con strumenti di modellazione commerciali e quindi *esportati* in un formato proprietario tramite plug-in appositamente creati.

Il sistema mantiene le relazioni di gerarchia definite dall'ambiente di modellazione, ed è in grado di gestire più oggetti correlati all'interno di un unico modello. Questa caratteristica è fondamentale per la creazione di animazioni complesse con l'utilizzo di pochi controlli.

Le relazioni gerarchiche permettono di definire le strutture ossee per l'animazione scheletrica, fondamentale per la creazione di personaggi virtuali; la flessibilità dell'animazione scheletrica è in grado di simulare adeguatamente i movimenti naturali della pelle e dei vestiti.

“Neo Euclide” supporta anche la *Shape-animation*. Anche questa tecnica permette di deformare dolcemente le geometrie, è computazionalmente meno costosa ma richiede un quantitativo di memoria elevato a causa della necessità di memorizzare molti dati per ogni chiave di animazione; risulta meno flessibile in quanto il numero di chiavi è necessariamente limitato. Solo in alcuni casi, la *Shape-Animation* si è rivelata essere la soluzione ottimale.

“Neo Euclide” gestendo entrambi i metodi da al modellatore la possibilità di stabilire quale utilizzare a seconda delle necessità.

Tra gli attributi gestibili tramite i sensori vi è anche la visibilità dei singoli oggetti o di intere gerarchie di oggetti all’interno della scena.

Formati dei File

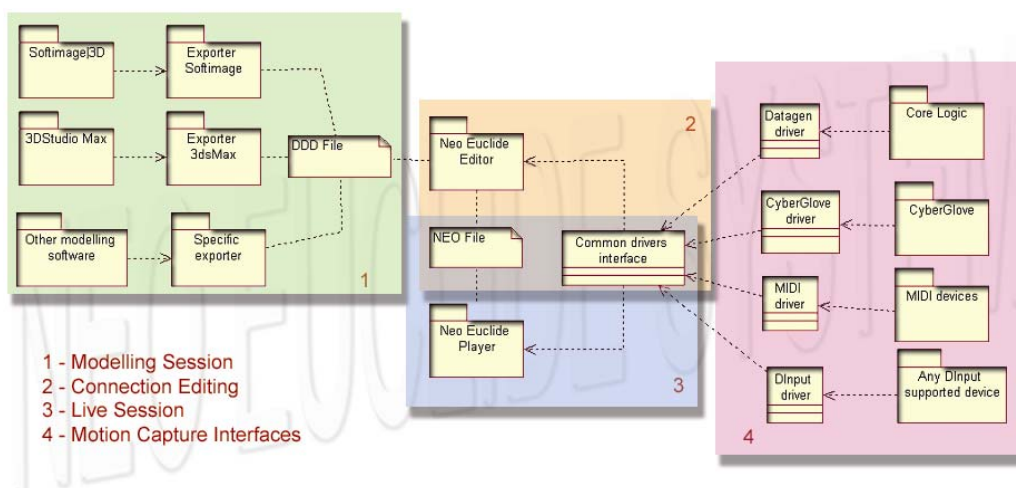
Tutti i parametri impostati all’interno del sistema vengono memorizzati in diversi file correlati tra loro, uno tipo per la scena 3D (.ddd), uno per l’audio (.wav) ed uno (.neo) per le relazioni tra i precedenti e i dispositivi di input. Per proteggere dalla pirateria e da usi non autorizzati il sistema, i file che contengono i modelli sono in un formato proprietario (.ddd).

L’uso di un formato proprietario permette l’ottimizzazione delle geometrie e la semplificazione di alcuni elementi in modo da renderli compatibili con il nostro motore di rendering. Il file esportato include quindi geometrie, relazioni di gerarchia, materiali, riferimenti per le texture, animazioni, relazioni con gerarchie scheletriche ed altro ancora.

Utilizzo

L’utilizzo di “Neo Euclide” è l’ultimo passo nella produzione di una marionetta virtuale. In questa fase vengono stabilite le connessioni ed il modello può cominciare a muoversi.

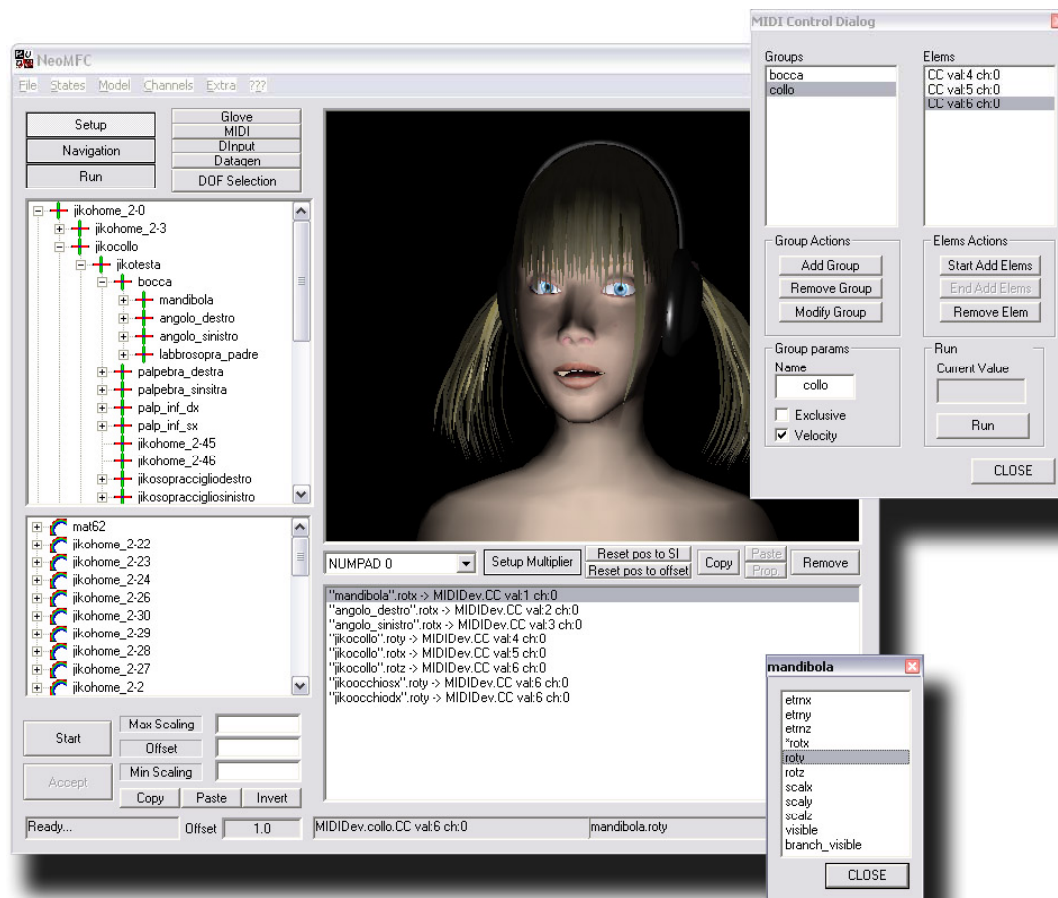
“Neo Euclide” è composto da un Editor per le connessioni dei dispositivi agli elementi della scena, ed un Player, necessario per l’animazione live. L’editor permette all’utente di connettere facilmente qualsiasi sensore (dai segnali MIDI ai sensori dei guanti) a qualsiasi elemento della scena per controllarne posizione, rotazioni, scalature, animazioni etc.. Il player viene usato per l’animazione live ed è in grado di caricare modelli in fase di esecuzione.



“Neo Euclide” Editor

L’editor comprende molte funzioni per il “collegamento” della marionetta. Il primo passo è caricare il modello nel formato .ddd. Il secondo è caricare un file .neo precedentemente realizzato in cui vi sono già relazioni tra sensori ed elementi della scena oppure crearne uno nuovo.

L’ultimo passo è il più interessante: una volta caricato il modello, l’interfaccia dell’editor mostra un sommario degli elementi della scena che possono essere assegnati ai sensori. L’interfaccia permette la connessione dei dispositivi a qualsiasi elemento della scena e la definizione di movimenti e limiti tramite una visione 3D interattiva.



Abbiamo deciso di supportare nel motion-capture anche dispositivi a basso costo, come i data-glove, e praticamente qualsiasi tipo di strumento MIDI. Ad ogni sessione di lavoro i dispositivi possono variare senza che sia necessario adeguare l’applicativo con modifiche o particolari impostazioni.

Sensori

Un sensore può controllare il movimento di un singolo oggetto 3D e la sua visibilità. Se connesso ad un materiale può controllarne la risposta all'illuminazione, la componente di luce ambientale, diffusa, speculare e la trasparenza. Se collegato ad un suono può controllarne, l'esecuzione, il volume, la frequenza, il pan ed il posizionamento all'interno della scena.



Qualsiasi sensore può controllare più di un elemento, ma due sensori non possono controllare lo stesso elemento. L'animatore è libero di modificare le connessioni fatte tra sensori ed elementi della scena anche successivamente.

Qualsiasi tipo di sensore può essere usato in qualsiasi tipo di connessione ed in qualsiasi scena 3D.

Ad ogni sensore vengono assegnati tre valori per la posizione: minimo, intermedio e massimo. La posizione di riposo di un dito non sempre corrisponde al valore medio del sensore che muove, e ancor più spesso i movimenti della mano dell'animatore non raggiungono i limiti estremi dei sensori. Abbiamo introdotto un fase di setup per i sensori dei dispositivi per evitare posizioni scomode o dolorose per l'animatore. Inoltre spesso è necessario mappare piccoli movimenti dell'animatore su grandi movimenti del modello, e a volte la posizione di riposo dell'animatore è vicina al valore massimo o minimo di alcuni sensori.

E' possibile usare un sensore per modificare l'intervallo di lavoro di un altro sensore. Quest'associazione viene chiamata *multiplier-link*.

Per esempio uno slider MIDI può influenzare l'interpretazione di un dato sensore, connesso alla rotazione di un oggetto, in questo modo possiamo stabilire che quando lo slider MIDI si trova alla posizione 0, il sensore è in grado di ruotare l'oggetto in un intervallo che va da -90 a 90 gradi, diversamente se lo slider MIDI si trova nella posizione 1 il sensore può ruotare l'oggetto in un intervallo da -180 a 180 gradi.

Raffinamento

A questo punto la “marionetta” è connessa; ma non ancora raffinata. Utilizzando la funzione “Navigator” e la funzione “Run” di “Neo Euclide Editor” l’animatore può controllare se ci sono problemi di connessione ed eventualmente ricalibrare un sensore. Inoltre l’animatore può accorgersi di eventuali errori all’interno della scena 3D (elementi non correttamente modellati o correlati tra loro etc.) o nell’animazione (gerarchie scheletriche che modificano erroneamente le geometrie etc.) oppure cercare di rendere più naturali i movimenti del modello virtuale.



“Neo Euclide” Player

Il player viene utilizzato durante l’animazione live; può funzionare all’interno di una finestra o occupare l’intero schermo dando pieno spazio alla fase di real-time rendering durante la performance.

Il player è in grado di caricare le scene 3D anche in fase di esecuzione; questo permette di effettuare cambi di scena totali in qualsiasi momento.



Utilizzo e prove

Abbiamo interfacciato “Neo Euclide” ad “Eyesweb” [7] – un applicativo per il trasferimento e la manipolazione dei dati – permettendo di controllare più stazioni di rendering da una singola stazione connessa ai dispositivi di motion-capture.

Stiamo lavorando anche all’interfacciamento tra “Neo Euclide” e “Lipsync” [8] – un software in grado di sintetizzare le espressioni del viso analizzando la voce – per l’automazione dei movimenti delle labbra.

Attualmente il sistema è in uso al museo “Città della Scienza” di Napoli [9] per animare la guida del museo: “Bit”. Questa installazione è stata creata da “Studio Azzurro S.R.L.” di Milano [10]; consiste in 5 postazioni – dove il personaggio virtuale interagisce con il pubblico – ed una stanza di controllo – dove l’animatore può decidere quale postazione animare.

La performance dell’animatore viene registrata e salvata all’interno di un database di eventi per essere utilizzata da un sistema di intelligenza artificiale. Questo sistema – realizzato dal DIST dell’Università di Genova – controlla automaticamente le locazioni non controllate dall’animatore [5,6].

Sviluppi futuri

Il sistema permette già il controllo remoto dell'ambiente 3D attraverso il protocollo TCP/IP, installando localmente i dispositivi di input e dedicando una macchina remota al real-time rendering. Sono previsti altri sviluppi per l'automazione del controllo di alcuni elementi, ad esempio l'integrazione con Lipsync, in grado di controllare l'animazione delle labbra di un volto virtuale, ed altri elementi che caratterizzano lo stato d'animo come ad esempio le sopracciglia, analizzando il suono della voce.

Conclusioni

Abbiamo presentato "Neo Euclide", un sistema per l'animazione live di scene tridimensionali dalle seguenti proprietà:

- supporta hardware a basso costo
- è generico
- permette elevate prestazioni
- rende possibile la comunicazione con altri sistemi

Il sistema, chiamato "Neo Euclide", è aperto, compatibile con svariati dispositivi di input e di acceleratori grafici, che permettono all'animatore di scegliere la configurazione migliore a seconda delle esigenze e dei limiti economici. Il sistema utilizza modelli realizzati con comuni ambienti di modellazione attraverso plug-in dedicati in grado di esportare le scene 3D in un formato proprietario. "Neo Euclide" può essere interfacciato attraverso il protocollo MIDI ad altri sistemi software o dispositivi di input.

Abbiamo introdotto un nuovo metodo per l'interpretazione dei dati provenienti dai guanti virtuali (posizione minima, di riposo e massima), ed un nuovo metodo per manipolare i dati provenienti dai sensori dei dispositivi MIDI. Il motore di rendering proprietario, "R.U.D.O." (Rendering Unit for DirectX Object) è compatibile con la maggior parte degli acceleratori grafici moderni.

Ringraziamenti

Un ringraziamento particolare al professor Giovanni Degli Antoni per l'entusiasmo trasmesso, per la disponibilità e per il supporto dato; a Davide Bertoncini, autore delle foto in questo articolo.

Appendice - Dispositivi di input

Abbiamo utilizzato diversi dispositivi di input, guanti, periferiche MIDI, e dispositivi standard come mouse e tastiere.

Guanti Virtuali (Data-Glove)

FiveTh Glove

E' composto da 8 sensori, 5 sono basati su fibra ottica e rilevano la chiusura di ogni dito con 8bit di precisione, 2 rilevano la rotazione del polso anch'essi con 8bit di precisione. Il FiveTh Glove è collegato tramite una porta seriale ed è alimentato a 9 volt con un alimentatore esterno. Per la gestione viene usato un driver per Windows che fornisce le interfacce necessarie tramite una libreria.

E' adatto ad animare gli elementi che compongono il viso di un personaggio virtuale come bocca, labbra, palpebre e sopracciglia, due sensori che campionano la rotazione del polso possono essere utilizzati per muovere la rotazione del collo.

Le fibre ottiche sono fragili ma precise, il numero di controlli è limitato, i due sensori per la rotazione del polso sono imprecisi e molto soggetti al rumore dato dalle piccole rotazioni causando fastidiosi effetti di tremolio. Questo guanto è molto economico.



Tecnologia	Fully-Enclosed Fiber Optic Bend
Numero sensori	5 (uno per dito)
Risoluzione	bits (256 posizioni/sensore) +/- 60° Pitch & Roll 8-
Interfaccia di Output	Porta seriale, RS-232 (3-wire) 19.2 kbaud (full duplex)
Sampling rate	200 hz per sensore
Tracking	Sensore di Pitch & Roll integrati
Routine di calibrazione	Aprire e chiudere la mano
Software	DLL per windows, Mouse Emulation, Raw Data Viewer, modello di mano poligonale con sorgente C++, KineMusica programma di output MIDI.
Alimentazione	9v DC

HumanGlove

E' composto da 20 sensori con precisione selezionabile (8 o 12 bit). E' in grado di raggiungere una precisione di rotazione di 0.4-0.6 gradi. La sensoristica prevede anche le articolazioni distali in modo indipendente. Anch'esso è collegato tramite la porta seriale gestita direttamente da "Neo Euclide".

E' adatto ad animare gli elementi che compongono il viso di un personaggio virtuale come bocca, labbra, palpebre e sopracciglia, due sensori che campionano la rotazione del polso possono essere utilizzati per muovere la rotazione del collo.

Il numero di sensori è sufficiente a gestire una discreto numero di parametri per l'animazione. Alcuni sensori (in particolare quello del pollice) risultano piuttosto fragili. Si colloca in una fascia di prezzo media.



Tecnologia	Convertitori A/D a 12bit o 8bit
Numero sensori	20
Risoluzione	0.4 o 0.6 gradi costanti su tutto l'intervallo di lavoro (convertitori A/D a 12bit)
Interfaccia di Output	Porta seriale, RS-232 38400 baud
Sampling rate	Circa 1 grado
Tracking	1% massimo di non linearità su tutto l'intervallo di lavoro
Routine di calibrazione	Muovere la mano in corrispondenza dei sensori
Software	Con il guanto viene fornito un pacchetto software per la visualizzazione tridimensionale di un modello di mano grafica denominato <i>Graphical Virtual Hand (GVH)</i> per piattaforme: Silicon Graphics IRIX, Linux, Windows NT, Windows 95
Alimentazione	9v DC

Cyber Glove

E' composto da 18 o 22 sensori, a seconda della versione, con una risoluzione di 0.5 gradi. Anch'esso viene collegato tramite la porta seriale. E' adatto ad animare gli elementi che compongono il viso di un personaggio virtuale come bocca, labbra, palpebre e sopracciglia, due sensori che campionano la rotazione del polso possono essere utilizzati per muovere la rotazione del collo.

Il CyberGlove è leggero, preciso e lineare. E' considerato lo stato dell'arte dei guanti attualmente in commercio per le proprietà e la possibilità di espansione: è possibile collegare strumenti aggiuntivi per il force feedback.

Si colloca in una fascia di prezzo molto alta.



Tecnologia	Sensori di flessione (Resistive)
Numero sensori	18 o 22
Risoluzione	0.5 gradi
Interfaccia di Output	Porta seriale, RS-232 115kbaud
Sampling rate	150 hz per sensore con dati non filtrati, 112 con dati filtrati
Tracking	0.6% massimo di non linearità su tutto l'intervallo di lavoro
Routine di calibrazione	Muovere la mano in corrispondenza dei sensori
Software	Con il guanto viene fornito un pacchetto software per la visualizzazione tridimensionale di un modello di mano grafica denominato <i>Virtual Hand</i> , ed il software di calibrazione
Alimentazione	9v DC

Periferiche midi

E' possibile utilizzare praticamente qualsiasi periferica midi come dispositivo di input, i test sono stati effettuati con Mixer e Tastiera.

Mixer midi

Il mixer midi da noi testato è composto da 16 slider con 7 bit di precisione collegato tramite la gameport della scheda audio SoundBlaster Live! installata.

Il mixer midi si è rivelato adatto per controllare intensità di luce, posizione delle luci e delle telecamere, frequenze delle animazioni prodotte dai datagen. Non è adatto per gestire animazioni che prevedono veloci cambi di posizione degli slider. Ad ogni modo mette a disposizione 16 sensori che possono essere comunque utilizzati per le animazioni quando il numero di sensori presenti in altri dispositivi (ad es. guanto) è scarso. E' un dispositivo di fascia molto bassa.

Tecnologia	Potenzimetri
Numero sensori	16
Risoluzione	7bit
Interfaccia di Output	Porta midi
Sampling rate	-
Tracking	lineare su tutto l'intervallo di lavoro
Routine di calibrazione	-
Software	-
Alimentazione	9v DC

Midi keyboard

Abbiamo effettuato alcuni test utilizzando due tastiere. La prima contenete 6 ottave, la seconda più contenuta con solo 3 ottave. I singoli tasti della tastiera sono i sensori, ed il loro numero è dipendente dal numero di tasti della tastiera. Inoltre è possibile usare controlli proporzionali come il pitch bend.

Per ogni tasto è possibile rilevare la pressione e la velocity con 7bit di precisione. Anche questi dispositivi sono collegati tramite la gameport della scheda audio SoundBlaster Live! installata.

Le tastiere midi sono adatte al controllo della visibilità degli oggetti all'interno della scena 3d e per l'attivazione di animazioni. E' un dispositivo di fascia molto bassa.

Tecnologia	Dipendente dal dispositivo
Numero sensori	Dipendente dal dispositivo
Risoluzione	7bit per velocity e pitch bend
Interfaccia di Output	Porta midi
Sampling rate	-
Tracking	lineare su tutto l'intervallo di lavoro
Routine di calibrazione	-
Software	-
Alimentazione	Dipendente dal dispositivo

Periferiche standard

Mouse

Il numero di sensori che definiscono questo dispositivo è dipendente dal costruttore. Il sistema è in grado di gestire fino a 3 assi e 8 pulsanti. Il mouse da noi testato era provvisto di 3 assi (x, y e z implementato dalla rotella) e 3 pulsanti, collegato tramite la porta Ps2. diversamente si possono utilizzare mouse connessi tramite la porta seriale o la porta usb.

Il mouse è risultato comodo per controllare movimenti come la rotazione degli occhi, lo spostamento di semplici oggetti, o lo spostamento del personaggio principale, usando una particolare funzione matematica in grado di semplificare il cambio di sensibilità in funzione della distanza dell'oggetto dalla telecamera. E' un dispositivo di fascia molto bassa.

Tecnologia	Encoder ottici microinterruttori
Numero sensori	Fino a 3 assi, e fino a 8 pulsanti
Risoluzione	Dipendente dal dispositivo
Interfaccia di Output	Porta Ps2, seriale o Usb
Sampling rate	Dipendente dal dispositivo
Tracking	lineare su tutto l'intervallo di lavoro
Routine di calibrazione	-
Software	-
Alimentazione	Diretta dal PC

Pc Keyboard

Una tastiera da PC è normalmente composta da 105 tasti che possono essere utilizzati, collegata tramite la porta tastiera del PC o tramite Usb. La tastiera è comoda per controllare la visibilità degli oggetti e per stabilire il set di connessioni attuale (è possibile farlo con i tasti del tastierino numerico). E' un dispositivo di fascia molto bassa.

Tecnologia	microinterruttori
Numero sensori	Normalmente 105
Risoluzione	-
Interfaccia di Output	Porta tastiera o Usb
Sampling rate	-
Tracking	-
Routine di calibrazione	-
Software	-
Alimentazione	Diretta dal PC

Joystick

Il numero di sensori che compone un dispositivo di questo tipo è molto variabile, dipendente dal costruttore. Può avere fino a 6 assi, fino a 32 pulsanti e può avere un dispositivo aggiuntivo ad otto direzioni.

Il joystick da noi testato conteneva 3 assi e 4 pulsanti. La risoluzione degli assi è fortemente dipendente dal costruttore. I joystick vengono connessi tramite la porta game delle schede audio, tramite la porta usb ed in alcuni casi tramite la porta seriale.

Si è rivelato utile per controllare solo alcuni movimenti temporanei, in quanto richiede che la mano resti ferma trattenendo il joystick in una determinata posizione, non permettendo di utilizzarlo come dispositivo extra. E' un dispositivo di fascia molto bassa.

Tecnologia	Dipendente dal costruttore
Numero sensori	Fino a 6 assi, 32 pulsanti ed 1 dispositivo a 8 direzioni (Pov)
Risoluzione	Dipendente dal costruttore
Interfaccia di Output	Porta game della scheda audio, Usb o seriale
Sampling rate	-
Tracking	-
Routine di calibrazione	Dipendente dal dispositivo, spesso interna al S.O.
Software	-
Alimentazione	Diretta dal PC

Riferimenti

1. David J. Sturman. Computer Puppetry. IEEE Computer graphics and application. January/February 1998.
2. M.Brand. Shadow Puppetry. *int. Conf. on Computer Vision, ICCV '99*, 1999.
3. V.Banquey and L.Juppé, The use of Real-Time Performance Animation in the Production Process. Motion Capture in Practice, ACM Siggraph 97 Course Notes, ACM Press, New York, 1997.
4. L.K. Saul and M.I. Jordan, A variation principle for model-based interpolation. Technical Report, MIT Center for Biological and Computational Learning, 1996.
5. A.Camurri, P.Ferrentino, *The Other Way - A Change of Viewpoint in Artificial Emotions* Proc. IEEE Intl. Conf. on Systems Man and Cybernetics SMC'98, San Diego.
6. K. Suzuki, A. Camurri, S. Hashimoto, P. Ferrentino, *Intelligent Agent System for Human-Robot Interaction through Artificial Emotion* Proc. IEEE Intl. Conf. on Systems Man and Cybernetics SMC'98, San Diego.
7. A.Camurri, P.Coletta, M.Peri, M.Ricchetti, A.Ricci, R.Trocca, G.Volpe, *A real-time platform for interactive performance*, Proc. ICMC-2000, Berlin
8. M. Malcangi, R. de Tintis, LipSync 1.0: A System for Real-Time Virtual Characters Lip-Synching and Facial Modeling, XIII Colloquium on Musical Informatics CIM 2000, pp 216-219, L'Aquila, 2000.
9. <http://www.cittadellascienza.it/>
10. <http://www.studioazzurro.com/>
11. Hyun Joon Shin, Jehee Lee, Sung Yong Shin, Michael Gleicher. Computer puppetry: An importance-based approach. ACM Transactions on Graphics (TOG) April 2001, Volume 20 Issue 2.
12. Francesca Barrientos. Continuous control of avatar gesture. Proceedings of the 2000 ACM workshops on Multimedia November 2000.
13. Nadia Magnenat Thalmann , Daniel Thalmann. Computer animation. ACM Computing Surveys (CSUR) March 1996, Volume 28 Issue 1.