

03063
8



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

**UN MODELO PARA EL CONTROL DEL
MOVIMIENTO HUMANO BASADO EN
CINEMÁTICA INVERSA**

T E S I S

QUE PARA OBTENER EL GRADO DE:

MAESTRO EN CIENCIAS

P R E S E N T A:

RICARDO DUARTE PÉREZ

DIRECTORA DE LA TESIS: MA. ANA LUISA SOLIS GONZALEZ COSIO

**TESIS CON
FALLA DE ORIGEN**



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

~~EXEMPLES NO DE LA~~
DE LA BIBLIOTECA

EXEMPLES NO DE LA
DE LA BIBLIOTECA

A mi Familia:

**mis Padres
mi Esposa
mi Suegra
mis Hermanas
mis Hermanos
mis Sobrinas
mis Sobrinos
mis Cuñadas
mis Cuñados**

**a la que considero mi tesoro
Por su apoyo incondicional.**

Agradecimientos

Quiero expresar un sincero agradecimiento a mi asesora de tesis Ana Luisa Solís por su gran experiencia y apoyos que me ha brindado para terminar este trabajo.

Un sincero agradecimiento para los miembros del jurado que revisaron el presente trabajo enriqueciéndolo con sus observaciones y puntos de vista:

Dr. JESUS SAVAGE CARMONA.

Dr. BORIS ESCALANTE RAMIREZ.

Dr. YU TANG XU.

Dr. HOMERO VLADIMIR RIOS FIGUEROA.

Un agradecimiento para todos mis profesores (as) por su trabajo de enseñanza y apoyo con sus experiencias.

Un agradecimiento para todos mis compañeros (as), amigos (as) por compartir horas de estudio, de clase y de convivencia que hacen más ameno el estudio.

Un agradecimiento para todo el personal administrativo por su gran labor de apoyo y trato humano hacia nosotros los estudiantes.

Y por último un agradecimiento al ININ por otorgarme el pago de tiempo durante las horas de clase, pues sin el no me hubiera sido posible el realizar estos estudios.

Índice.

Introducción.	1
1. Representación de Humanos Virtuales o Avatares.	4
1.1 Antecedentes.	4
1.2 Representación de Avatares.	7
1.3 Avatares dentro del NVE.	9
1.4 Arquitectura para Avatares en NVEs.	9
1.5 Opciones de Diseño para una Representación	11
1.6 Opciones de Diseño para Control.	13
1.7 Aplicar Control al Movimiento.	16
1.8 Actores Perceptivos Autónomos e Interactivos	22
2. Generación del Modelo Humano.	30
2.1 Antecedentes	30
2.2 Modelado Geométrico.	31
2.2.1 Modelos basados en Superficies	31
2.2.2 Modelos de Volumen y CSG	33
2.3 Representando Figuras Articuladas	35
2.4 Modelo del Esqueleto Humano	36
2.4.1 Representación del Esqueleto	36
2.4.2 Jerarquías.	37
2.5 Crear el Esqueleto Humano.	38
2.5.1 Restricciones.	38
2.5.2 Condiciones Iniciales.	38
3. Análisis del Caminar Humano.	46
3.1 Antecedentes	46
3.2 Movimiento	50
3.2.1 Caminar Humano.	53
3.2.2 Proceso de Caminar.	55
3.3 La Raíz de la Figura.	56
3.4 Caminado básico Paso a Paso	57
3.4.1 La Cabeza y la Espalda.	57
3.4.2 Las Caderas, Pelvis, Torso, Espalda, Hombros	57

3.4.3 Los Brazos.	62
3.4.4 Las Rodillas y los Codos.	66
3.4.5 Los Pies y las Piernas	67
3.5 Ciclo de Caminar Humano.	70
4. Métodos de Cinemática y Dinámica para la Especificación del Movimiento.	76
4.1 Métodos de la Cinemática.	76
4.1.1 Antecedentes	77
4.1.2 Cinemática Directa	78
4.1.3 Cinemática Inversa.	79
4.2 Métodos Dinámicos	79
4.2.1 Antecedentes	80
4.2.2 Dinámica Directa	81
4.2.3 Dinámica Inversa.	82
4.2.3.1 Restricciones Geométricas.	83
4.2.3.2 Sin Restricciones Geométricas.	84
4.3 Controles del Movimiento.	84
4.3.1 Antecedentes	84
4.3.2 Control del Movimiento Cinemático y Dinámico.	85
5. Diseño y Control del Caminar de la Figura Articulada.	87
5.1 Antecedentes.	87
5.1.1 Condiciones de la Figura Articulada y Robótica en la Animación	88
5.2 El Problema de Cinemática Inversa.	89
5.2.1 Cinemática Directa	90
5.2.2 Cinemática Inversa.	91
5.3 Resolver la Velocidad para Controlar el Movimiento.	92
5.3.1 El Jacobiano.	93
5.3.2 Construyendo el Jacobiano	94
5.3.3 Invirtiendo el Jacobiano	94
5.4 Mapear Cambios.	95
5.5 Un Paso Iterativo hacia la Meta.	95
5.6 Redundancia.	97
5.7 Las Singularidades.	98
5.8 Condicionamiento.	100
5.9 Métodos basados en Optimización.	102

6. Animación. El Caminar de la Figura Humana.	104
6.1 Antecedentes	104
6.2 Esqueleto Humano, sus Ejes Coordinados y la Animación	105
6.3 Parámetros del Movimiento.	115
6.4 Cinemática Directa Aplicada a los Esqueletos	116
6.5 Cinemática Inversa Aplicada a los Esqueletos	119
6.6 Cinemática Inversa en Brazos, Piernas y Pies.	121
6.6.1 Expresión Analítica del Brazo.	122
6.6.2 Expresión Analítica de las Piernas.	124
6.6.3 Cinemática Inversa del Pie	127
6.7 Explicación del Movimiento.	128
6.8 Animación	129
7. Un Análisis mas avanzado: Generación automática de pasos.	132
7.1 Antecedentes	132
7.2 Modelo de la Cinemática	133
7.3 ElevWalker: La generación del movimiento de pasos	134
7.4 ElevInterp: Transformación de datos del movimiento	138
7.4.1 Interpolación del movimiento inverso	138
7.5 MetaGait	140
7.5.1 Curva de locomoción.	141
7.5.2 Locomoción en terreno desigual.	143
7.5.3 Determinando los nuevos valores del parámetro	144
7.6 Resultados.	144
7.7 Discusión y Trabajo Futuro	146
Conclusiones.	147
Anexo A Diagrama de Bloques.	149
Anexo B Diagrama de Flujo y Módulos.	150
Anexo C Programa de Animación.	160
Bibliografía.	184

Introducción.

Los sistemas interactivos en computadoras con capacidades gráficas han existido desde principio de los setentas apoyando el modelado, manipulación y animación de figuras humanas.

Actualmente una de las áreas de investigación en la interacción entre hombre-computadora y en el área de graficación por computadora, es la creación de ambientes virtuales inteligentes donde puedan incorporarse agentes que interaccionen con el humano.

Esto implica el desarrollar humanos virtuales donde su comportamiento y movimiento sean creíbles. Para poder incorporarlos es necesario el estudiar y modelar su movimiento. Aunque la animación de la figura origina los desafíos en el modelado y el rendering, el problema fundamental de diseñar y controlar el movimiento de estas figuras, actualmente es uno de los más difíciles de realizar.

Los movimientos de las partes individuales del cuerpo deben conocerse en cada momento y coordinándose entre ellas.

Por eso se busca construir un modelo computacional de una figura humana que genere una animación convincente. Dada la estructura compleja de una figura articulada típica, puede involucrar una gran cantidad de trabajo.

Existen varios métodos para resolver el problema, pero el objetivo de este trabajo es usar el método de la cinemática inversa para resolverlo. Para estos fines

- Crear un modelo humano (diseñar la figura o esqueleto)

- Analizar, diseñar y controlar el movimiento de la figura.
- Aplicar la cinemática inversa en brazos, piernas y pies para generar su movimiento.
- Generar la animación de la figura.

La presente tecnología puede aproximar la apariencia humana del caminado a través del modelo de un esqueleto humano (figura articulada) en computadoras con capacidades gráficas y animación en 3D usando cinemática.

Actualmente no hay soluciones perfectas a ninguno de estos problemas, pero avances significativos han habilitado la consideración de las metas adecuadas bajo suposiciones uniformes y consistentes.

Algunas aplicaciones de los modelos de figuras humanas en 3D son: simulación de caída, análisis de movimiento, estimación del lugar de trabajo, notación de movimiento o baile, entretenimiento, entender el movimiento, medicina (rehabilitación), entrenamiento deportivo, estudios de ergonomía (diseño automotriz, etc)

El deseo en éste trabajo es el de generar un modelo de una figura que se adapte a las técnicas de cinemática para generar la animación del movimiento del modelo. Se repasan los problemas: representación de humanos virtuales, generación del modelo humano, entendimientos del movimiento, cinemática, dinámica y sus soluciones que se han aplicado anteriormente en la animación de una figura, el jacobiano y su inversa usando un método interactivo, así como un análisis de la generación automática de pasos.

Los algoritmos de cinemática inversa aplicados en la animación del modelo o figura articulada son sencillos y eficaces para producir una animación convincente.

Hay varias contribuciones de este trabajo algunas son el análisis y la descripción del movimiento, sus restricciones, pero la contribución principal es el análisis comparativo del cuerpo humano con el modelo de la figura articulada y así generar el movimiento de la parte baja del cuerpo y el movimiento para el cuerpo superior mejorando el realismo de la animación del caminado.

Organización.

En el capítulo 1, se explica la representación de los humanos virtuales en los ambientes virtuales.

En el capítulo 2, se explican los diferentes tipos de modelos y su representación, el modelo del esqueleto humano, su creación y representación

En el capítulo 3, se describen los componentes del cuerpo humano y se describen los movimientos de cada componente para describir el caminar humano paso a paso.

En el capítulo 4, describe los métodos de cinemática, dinámica, restricciones y sus controles de movimiento.

En el capítulo 5, se describe el problema de cinemática inversa, la solución para controlar el movimiento, incluyendo redundancia, singularidades, condicionamiento y métodos basados en optimización.

En el capítulo 6, se describe la representación del cuerpo humano, sus ejes coordenados, su animación, sus parámetros de movimiento, la cinemática inversa aplicada a esqueletos, la cinemática inversa aplicada en brazos, piernas y pies, su movimiento, animación y resultados obtenidos

En el capítulo 7, se realiza un análisis de los componentes para la generación automática de pasos.

Finalmente se encuentran algunas conclusiones basadas en los resultados obtenidos del algoritmo para la animación del caminar humano



Capítulo 1.

Representación de Humanos Virtuales o Avatares.

Uno de los desafíos de la investigación particularmente importante en ambientes virtuales (VEs), es la representación de avatares, que es la manera en que se representa a los participantes gráficamente en un ambiente virtual (VE). Su representación como humanos virtuales es muy realista con cuerpos articulados y caras. En tiempo real, los avatares serán esenciales en el futuro, cuando una buena representación del usuario pueda mejorar la presencia de un usuario en el ambiente y su habilidad de comunicarse con otros usuarios en ese sentido. También necesitamos los humanos virtuales autónomos para poblar los mundos virtuales. El objetivo es explicar la representación de los humanos virtuales en los ambientes virtuales. El material descrito se basa en las referencias [Igo99].

1.1 Antecedentes.

Ambientes Virtuales.

Ivan Sutherland introduce el concepto de insertar a personas en mundos generados en la computadora en 1965. Desde entonces, junto con el desarrollo del conocimiento de la generación de imágenes por computadora y tecnología, se han desarrollado varios sistemas que contienen ambientes virtuales. Steve Ellis ha presentado una introducción importante a los conceptos de ambiente virtuales (Ellis 1991). Define la virtualización como “el proceso por el cual un espectador humano interpreta un patrón de impresión sensorial para que un objeto sea extendido en un ambiente en el que exista físicamente”. La virtualización considera tres niveles: el espacio virtual, la imagen virtual y el ambiente virtual. Un ejemplo del espacio virtual, es la descripción de una escena y el observador puede visualizar los objetos tridimensionales a través de este espacio. La imagen virtual se refiere a la percepción del objeto con profundidad. El ambiente virtual, incluye al participante como la parte del mundo virtual, para que (él / ella) desde su punto de vista y acciones en este mundo correspondan a aquéllos del mundo físico (real)

En un ambiente virtual, los aspectos sensoriales se incorporan a los sentidos del participante a través de despliegues generados en computadora. Un ambiente virtual puede proporcionar tres

elementos para incorporar a un participante en el ambiente: el aspecto visual, el auditivo y el de sensación de tacto (haptics). El ambiente virtual es una intersección de estos elementos. Esta intersección es importante porque muestra un ambiente totalmente interactivo incluyendo todos sus elementos. Sin embargo, en esta definición los ambientes virtuales generales y más pequeños pueden ser creados usando subconjuntos de los tres componentes. Kalawsky (1993) definió los ambientes virtuales como “experiencias sensoriales sintéticas que comunican a los componentes físicos y abstractos de un participante. La experiencia sensorial sintética se genera por un sistema de computadora que puede presentar una interfaz a los sistemas sensoriales humanos que son indistinguibles del mundo físico al real”. El ambiente virtual se refiere a una tecnología que es capaz de cambiar un sujeto a un ambiente diferente sin moverlo físicamente. Con este fin, se manipulan las entradas en los órganos sensoriales del sujeto, de una manera tal que el ambiente percibido es asociado con el ambiente virtual deseado, no con el ambiente físico. El proceso de manipulación se controla por un modelo en una computadora que se basa en la descripción física del ambiente virtual. Por consiguiente, la tecnología puede crear casi arbitrariamente los ambientes percibidos. Hay otras definiciones de un ambiente virtual. Sin embargo, la mayoría de las definiciones están de acuerdo en apoyar a la totalidad de los sentidos involucrando al participante en el ambiente virtual.

Ambientes Virtuales Distribuidos.

Recientemente, la mayoría de los sistemas basados en mundos virtuales han sido de solo un usuario. (Singh y Serra 1994). Compararon estos con los sistemas tradicionales, los sistemas de solo un usuario proporcionan un tremendo beneficio al usuario pero su utilidad en el mundo real está bastante limitada. Esto es porque ellos no apoyan la colaboración entre un grupo de usuarios. Una red de computadoras acopladas a tecnología altamente interactiva de mundos virtuales dominará el mundo de computadoras y la tecnología de información (figura 1.1) [Igo99]. No será suficiente el producir a un solo-usuario, en los mundos virtuales autónomos.

Estos sistemas tendrán que conectar a las personas, a los sistemas, al flujo de información y a tecnologías entre sí. La información que es actualmente compartida por medio de sistemas de archivos o de otros medios de comunicación, tendrán que ser intercambiadas a través de la información de redes de trabajo. Esta información tiene que residir en la red donde esta sea fácil de obtener. El desarrollo de mundos virtuales para apoyar la colaboración entre un grupo de usuarios es una tarea compleja y consume tiempo. Para desarrollar tales mundos virtuales, el diseñador tiene que ser hábil programando en la red, en la dirección del objeto, en la programación gráfica, en el control de dispositivos y en el diseño de la interfaz del usuario. Las tendencias hacia las aplicaciones en red de computadoras y apoyando el trabajo colaborativo en computadoras, junto con un interés amplio para los sistemas gráficos y los ambientes virtuales, en los años recientes levantaron el interés por la investigación en el campo de ambientes virtuales distribuidos (NVEs) (Durlach y Mavor 1995). NVEs son sistemas que permiten usuarios múltiples geográficamente distantes para interactuar en un ambiente virtual común. Los mismos usuarios que utilizan una personificación gráfica se representan dentro del ambiente (figura 1.2) [Igo99].

El usuario puede evolucionar dentro del ambiente y puede interactuar con él. Todos los eventos que tienen un impacto en el ambiente se transmiten a otros sitios para que todos los ambientes puedan ser puestos al día y guarden consistencia, dando la impresión para los usuarios de estar en el mismo

y único ambiente. Los usuarios llegan a ser una parte del ambiente, incluyendo una representación gráfica que debe ser parecida al humano.

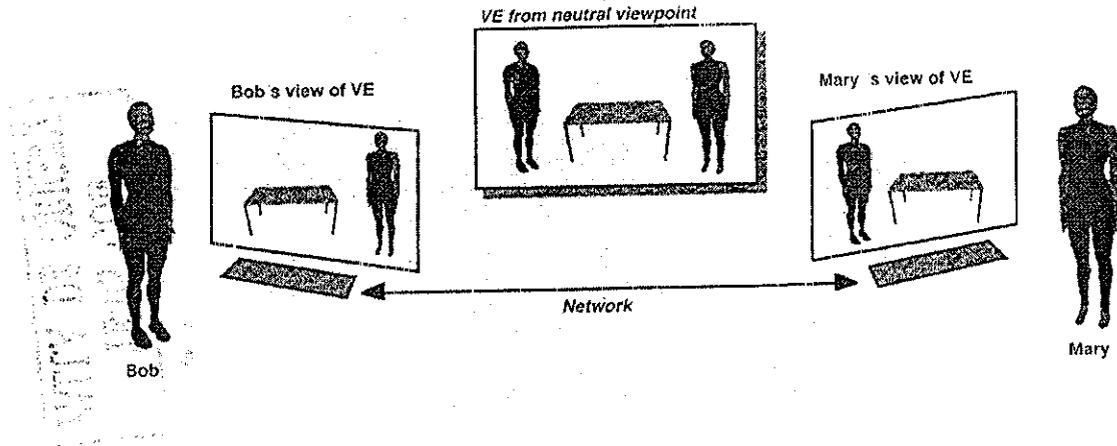
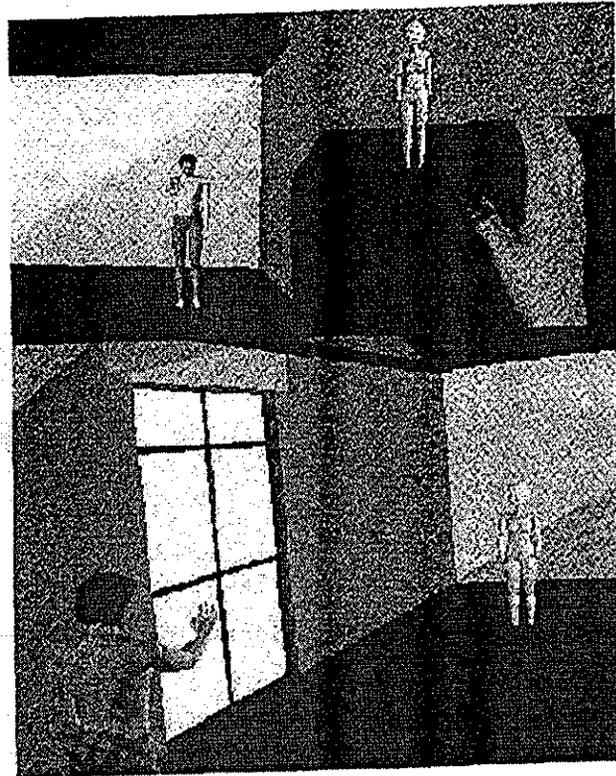


Figura 1.1. Principios del ambiente virtual distribuido.

Los sistemas de ambientes virtuales distribuidos son adecuados para numerosas aplicaciones colaborativas que van de los juegos a la medicina (Doenges et al. 1997; Zyda y Sheehan 1997), se tienen los siguientes ejemplos:

- La teleconferencia virtual con el intercambio de información multimedia
- Trabajo colaborativo que involucra diseño 3D.
- Los ambientes de juego multi-usuario.
- Telecompras que involucra a modelos 3D, las imágenes, el sonido (por ejemplo bienes raíces, el mobiliario, los automóviles).
- Las aplicaciones médicas (los diagnósticos a distancia, cirugía virtual para entrenamiento).
- Aprendiendo / entrenando a distancia
- El estudio virtual / el juego con integración de medios de comunicación en ambientes distribuidos.



TESIS CON
FALLA DE ORIGEN

Figura 1.2. (arriba) dos vistas individuales, y (abajo) una vista total de la escena.

1.2 La Representación de Avatares.

Avatares.

Los avatares son un dominio de la investigación preocupada en la simulación de seres humanos en computadoras. Involucra representación, movimiento y conducta. Hay una gama amplia de aplicaciones (figura 1.3) [Igo99]: películas y producciones de televisión, en estudios ergonómicos y de usabilidad en varias industrias como aeroespacial, automóvil, maquinaria, mobiliario, etc., simulaciones de línea de producción para estudiar su eficacia, industria del vestido, juegos, las telecomunicaciones, medicina, etc. Estas aplicaciones tienen varias necesidades. Una aplicación médica podría requerir una simulación exacta de ciertos órganos internos; la industria cinematográfica requiere de las normas estéticas más altas, de movimientos naturales y de expresiones faciales; estudios ergonómicos requieren proporciones fieles del cuerpo para un segmento de población particular y del movimiento real con restricciones

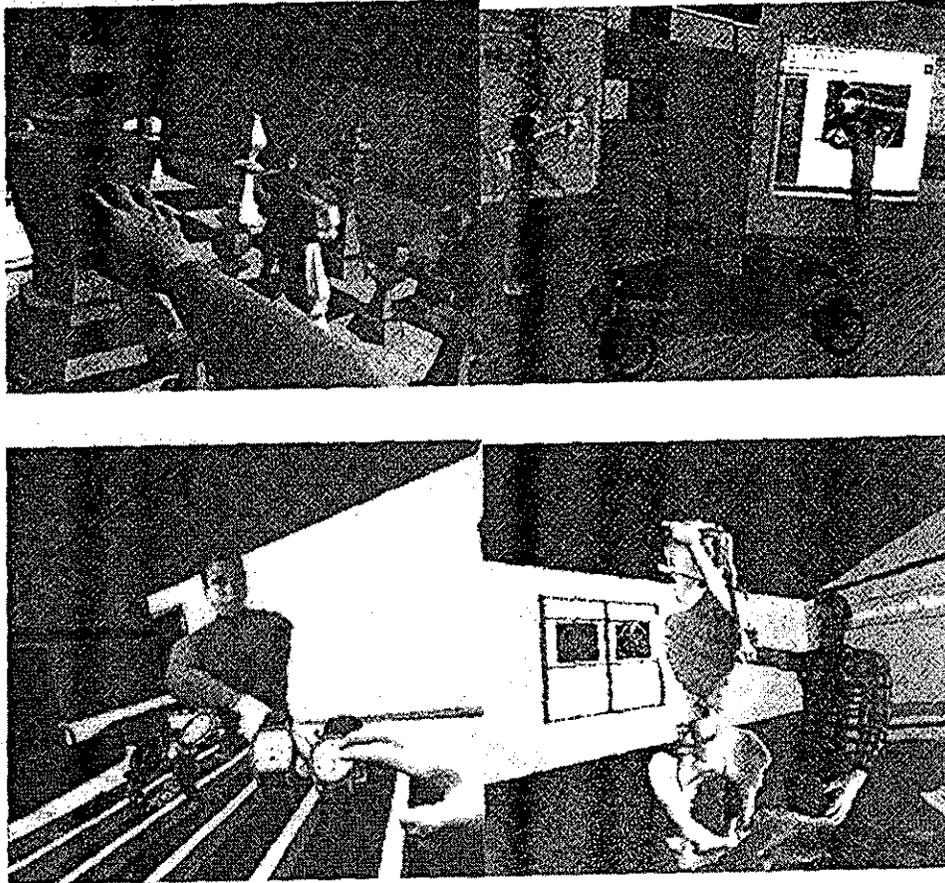


Figura 1.3. aplicaciones de vida real distribuida.

La pura complejidad del sujeto simulado, es decir el cuerpo humano, combinado con la multitud de aplicaciones y requisitos, hace a los avatares un dominio de la investigación inmenso que comprende numerosos temas de la investigación:

- La anatomía y geometría, tratan la creación de forma humana en 3D, con métodos que digitalizan modelos de arcilla punto a punto a través de herramientas del software para la deformación de la geometría y modelado, a scanners láser 3D
- La representación del pelo, la piel y su descripción
- La animación del esqueleto, o animación de ángulos de unión en la estructura de un esqueleto definiendo el cuerpo articulado y consistiendo de segmentos (representando los miembros) y uniones (representando grados de libertad) Los métodos principales de animación del esqueleto son la animación paramétrica del keyframe, las cinemáticas directas e inversas (con restricciones), la dinámica directa e inversa
- La animación del cuerpo y su deformación, intentando simular el movimiento de apariencia natural y la deformación visible del cuerpo con respecto al movimiento en la estructura del esqueleto.
- La animación y deformación de la mano

- La animación facial, juega un papel esencial en la comunicación humana. Hay dos áreas de la corriente principal de investigación en la animación facial: modelos parametrizados y modelos del músculo.
- La simulación del vestir o ropa.
- Caminado, es decir, genera el movimiento del caminado en apariencia natural, basado en una trayectoria y velocidad dada.
- La anulación de obstáculos, encontrando la trayectoria óptima, por haber caminado evitando obstáculos.
- Simulando movimientos apropiados del brazo y su mano para alcanzar y agarrar un objeto.
- La animación de conductas, esforzándose más por dar carácter y personalidad a la animación, haciéndola parecer más natural que las animaciones basadas en mecanismos.

1.3 Avatares dentro del NVE

La representación participante en un sistema NVE tiene varias funciones:

La percepción, la localización, la identificación, el enfoque de interés de visualización, la visualización de acciones, la comunicación.

La percepción y localización son funciones básicas que permiten percibir la presencia de otros en el ambiente y ver donde están. Incluso un avatar básico puede realizar estas tareas.

La identificación es una función que desea saber quién está a la vista. Los medios de identificación van de simples a complejos (como modelos de cuerpo y cara).

El enfoque de interés de visualización puede lograrse por cualquier avatar que de algún modo represente la dirección de su mirada - normalmente por medio de un modelo gráfico que tiene ojos o símbolos que representan los ojos, para que pueda ver en que dirección éste está mirando.

La visualización de acciones exige al avatar que tenga un resultado final que realice acciones. Una aplicación final podría ser una línea simple que alcanza al objeto, o podría ser una mano virtual que agarre el objeto.

La comunicación está ligada de muchas maneras a los gestos del cuerpo, a las expresiones faciales y son parte de nuestra comunicación diaria.

Aunque muchas de estas funciones pueden cumplirse con avatares simples, algunas funciones no pueden cumplirse sin estos. Los avatares pueden realizar estas funciones de manera intuitiva y natural, en forma parecida a tareas que realizamos los humanos en la vida real.

1.4 Arquitectura para Avatares en NVEs.

Incluir avatares en un ambiente virtual distribuido es una tarea compleja que combina varios campos de especialización:

- **La simulación del avatar**, involucra animación / deformación de cuerpos y caras en tiempo real y algunos otros desafíos
- **La simulación del ambiente virtual**, involucra la administración de bases de datos visuales y técnicas de descripción de imágenes en pantalla con optimizaciones en tiempo real.

- **Una red**, involucra la comunicación entre varios tipos de datos con varias necesidades en términos de rangos de bit y resaltando errores.
- **La interacción**, involucra el apoyo de diferentes dispositivos y modelos
- **La Inteligencia artificial** (avatares autónomos), involucra procesos de toma de decisión y conductas autónomas

Cada uno de los componentes involucrados representa completamente una área de investigación y la mayoría de estas son muy complejas. Al combinarlos, es esencial considerar la interacción entre los componentes y su impacto sobre cada componente. Esto tiene que permitir no sólo la interacción con el ambiente, si no también la visualización de acciones a través del cuerpo y la cara, al representar al usuario, la necesidad de comunicar los datos a través de la red, fuerza a los datos de animación a una representación más compacta de la cara y el cuerpo.

Debido a la complejidad de estos componentes, su tarea compleja se divide en módulos, cada uno con una función precisa y con las interfaces bien definidas entre módulos, se logran varias ventajas:

- Alta flexibilidad.
- Administración fácil del software, sobre todo en un ambiente de trabajo en equipo
- Rendimiento más alto
- Multiprocesadores o ambientes distribuidos cuando son disponibles

La flexibilidad es particularmente importante, debido a tecnologías de hardware y software que pueden unirse potencialmente con los sistemas de NVE (varios dispositivos de entrada y técnicas, algoritmos AI, fuentes de datos de tiempo reales que manejan las aplicaciones multiusuario) Esto es especialmente interesante en un ambiente de investigación dónde un sistema de NVE puede usarse en los campos de investigación como AI, la psicología, los sistemas de información médicos. En general, un buen sistema NVE debe permitir la implementación de diferentes aplicaciones mientras realiza sus tareas básicas transparentemente (red, representación del usuario, la interacción, descripción de imágenes, etc) y permite al programador de la aplicación, concentrarse en los problemas específicos de la aplicación.

Desde el punto de vista de **la administración del software**, un sistema monolítico de esta complejidad sería sumamente difícil de administrar, en particular por el equipo de programadores.

Puede lograr un rendimiento más alto, asignando las tareas cuidadosamente a los procesos y sincronizándolos inteligentemente.

Finalmente, **un sistema de multiprocesos** utilizará multiprocesadores si están disponibles. También es posible distribuir los módulos en varias redes.

Una vez que se decide dividir el sistema en módulos, es necesario definir la tarea de cada módulo y los medios de comunicación entre ellos.

1.5 Opciones de Diseño para una Representación.

Estructura Articulada para Modelar el Avatar.

Típicamente, se necesita una estructura articulada que represente al esqueleto humano para el control de la postura del cuerpo. Estructuras que representan la forma del cuerpo tienen que ser ligadas al esqueleto y pueden cubrir de ropa la forma de su cuerpo. El modelo usado del cuerpo humano articulado cuenta con 22 grados de libertad sin las manos. El esqueleto, mostrado en la figura 1.0, se representa por una jerarquía articulada 3D de uniones, cada una con los límites máximos y mínimos realistas. La postura predefinida es la figura 1.0. Note que la postura del cuerpo se representa por una serie de 22 valores de punto flotantes, cada uno representa el valor de un grado de libertad en el cuerpo. Hay conjuntos de parámetros diferentes para definir las posturas de avatares.

Los Parámetros del Dominio de Posicionamiento Global.

Éstos son la posición y orientación global, valores de puntos observables particulares en el cuerpo, dentro del sistema de coordenadas. Los puntos son la cabeza, el cuello, los hombros, el codo, la muñeca, la cadera, la pelvis, la rodilla, el tobillo.

Parámetros de Dominio de los ángulos de Unión.

Estos parámetros comprenden los ángulos de unión que conectan las diferentes partes del cuerpo.

La deformación del cuerpo es optativa: el cuerpo también puede definirse por partes sólidas del cuerpo ligadas al esqueleto.

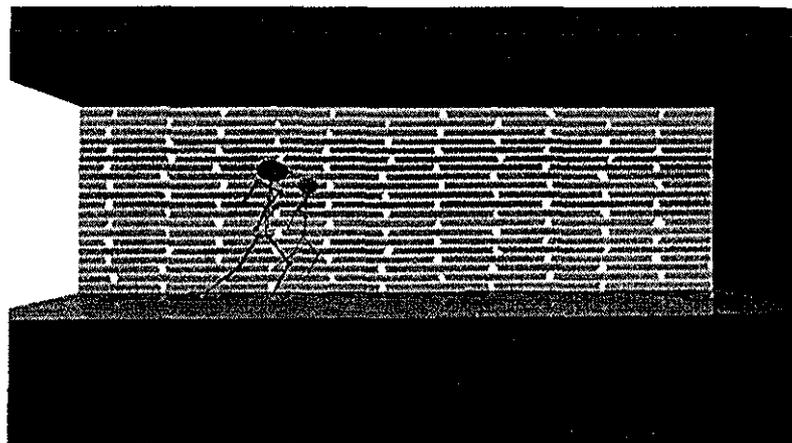


Figura 1.0. Modelo del cuerpo humano articulado cuenta con 22 grados de libertad.

Así, definir un espacio de estados de todas las posibles configuraciones del cuerpo articulado para un vector de espacios. El conjunto de parámetros independientes que definen las posiciones y orientaciones de todas las partes del cuerpo, forma la base del espacio de estados. Así definimos el cuerpo con una configuración que se describe por un vector de estados

$$v = (v_1, v_2, v_3, \dots, v_n)$$

Donde $v_1, v_2, v_3, \dots, v_n$ son los grados de libertad individuales en el cuerpo y la dimensión del vector es igual a los grados totales de libertad. La dimensión de nuestro modelo es de 22 sin las movibilidades de la mano

Escala para la Representación Virtual.

El problema para representar avatares en niveles múltiples de complejidad tiene dimensiones de diferentes escalas:

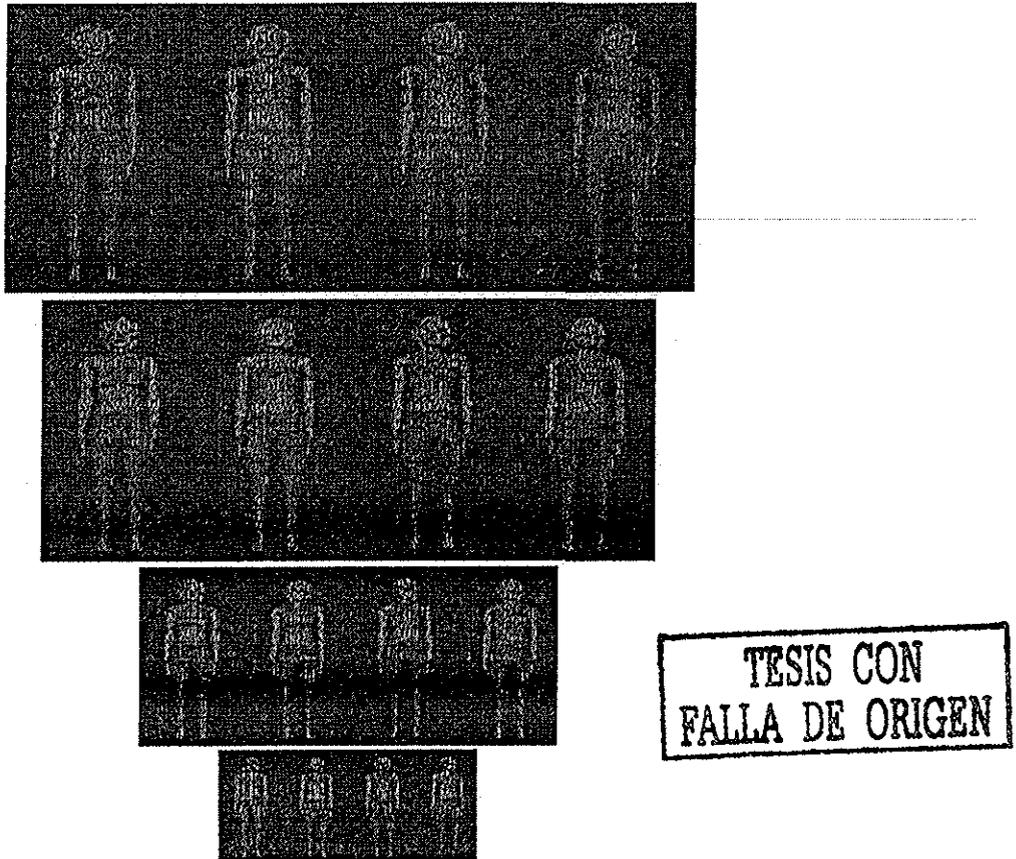


Figura 1-4. Niveles de detalle y resultados visuales vistos desde 4 distancias.

- **Escala de Procesamiento** se refiere al esfuerzo de trabajo local del avatar en la computadora central. Este tipo de escala es dividido en diferentes subtipos:

Escala Computacional: El esfuerzo de trabajo computacional que se necesita para poner al día, un conjunto de parámetros de la postura del cuerpo, para su animación. Este esfuerzo de trabajo es proporcional al tipo de parámetro de animación y al número de grados de libertad manipulados

Escala en gráficas: La cantidad de gráficas que necesita desplegar un cuerpo en tiempo real (figura 1-4) [Igo99]

- **Escala en red** se refiere al esfuerzo de trabajo en red de un avatar. Y sus tipos de escala son divididos en tres elementos:

Escala de Carga: La cantidad (de cálculo computacional y de red en esfuerzo de trabajo) para transmitir un modelo del cuerpo y cargarlo en memoria, durante la fase de la inicialización del nuevo participante.

Escala del ancho de banda: La cantidad de tráfico (se refiere a los bits por segundo) necesario para comunicar el estado del cuerpo en tiempo real.

Escala en codificación: El número de cálculos necesarios para poner en código y descifrar el mensaje transferido, a una lista de parámetros de animación del cuerpo, preparado para animar la copia remota de la representación de un cliente remoto.

- **Escala Computacional.**

La escala computacional define el esfuerzo de trabajo computacional para poner al día la postura del cuerpo virtual de un conjunto de parámetros de animación. Si la animación define que pone directamente al día los grados de libertad individuales, esta complejidad disminuye el recorrido en la jerarquía del cuerpo y pone al día la transformación de cada unión.

- **Escala en gráficas.**

En nivel gráfico el concepto de detalle se ha usado en varios paquetes de descripción de imágenes 3D y lenguajes como IRIS PERFORMER y VRML. Para cada avatar, han usado cuatro niveles de detalle (para optimizar algunas partes del sistema: CPU, Memoria, Red). El rango es normalmente muy dependiente del desplegado de pantalla o de las características de la ventana, la iluminación en la escena, la complejidad de VE, etc. La (figura 1-4) [Igo99] muestra los diferentes niveles de detalle para algunos cuerpos

1.6 Opciones de Diseño para Control.

Asociar e integrar a los participantes y a los actores autónomos con el sistema dentro del mundo virtual. Y describir sus **tareas principales para controlar el avatar** como son: **las tareas de percepción** (por ejemplo la identificación del ambiente), **las tareas cognoscitivas** (por ejemplo procesando la información y tomando decisiones), **las tareas de comunicación** (por ejemplo los gestos y otros medios de comunicación) y **tareas motoras** (por ejemplo como caminar, agarrar un objeto).

Tipos de Control para el Avatar.

El participante debe animar su representación del avatar en tiempo real; sin embargo, el control humano no es completo: la complejidad en la representación del avatar necesita rastrear un número grande de grados de libertad. Además, la interacción con el ambiente aumenta esta dificultad aún más. Por consiguiente, el control humano debe usar los mecanismos de más alto nivel, para poder animar su representación con una máxima facilidad y una mínima entrada. Se dividen los métodos para controlar el avatar en tres clases:

- **Control directo del avatar:** el vector de estados del avatar se modifica directamente (por ejemplo, usando sensores ligados al cuerpo) proporcionando un nuevo DOF directamente (por ejemplo, valores a los sensores ligados al cuerpo).

- **El usuario guía avatares:** el driver externo guía al avatar, definiendo las tareas a realizar, sus usos, sus habilidades de motor para realizar esta acción y los movimientos coordinados de la unión (por ejemplo camine, siéntese)

- **Avatares autónomos:** se asume que el avatar tiene un estado interior que se construye por sus metas e información del sensor del ambiente y el participante modifica este estado definiendo motivaciones de alto nivel y cambios de estado (por ejemplo activar la conducta de visión)

Los métodos de control no son independientes, los controles de más alto nivel requieren capacidades de más bajo nivel. La conducta autónoma asume las habilidades de motor para lograr el control y para modificar DOFs individuales.

Avatares Controlados Directamente.

Se requiere que el avatar tenga un cuerpo de apariencia natural y este se anime con respecto al cuerpo real. Esto corresponde a una forma de tiempo real en el rotoscopy tradicional. El rotoscopy tradicional en la animación consiste en la grabación del movimiento por un dispositivo específico para cada escena o foto (frame) y usa esta información para generar la imagen en la computadora. Usando la terminología introducida por Thalmann (1993), lo llamamos método de rotoscopy de tiempo real, a un método que consiste de datos de entrada magnetofónicos a un dispositivo de VR en tiempo real, que nos permite aplicar los mismos datos, a un objeto gráfico sobre la pantalla. Por ejemplo, cuando el animador abre los dedos 3 centímetros, la mano en la escena virtual hace exactamente lo mismo. Además, el registro de keyframes grabados previamente requiere de entradas en tiempo real de la geometría para una postura del cuerpo. La geometría de entrada puede darse, como los parámetros de posicionamiento globales o parámetros de ángulo de unión.

Una representación completa del cuerpo virtual participante, debe tener los mismos movimientos que el cuerpo participante real, para una interacción más inmersa. Esto puede lograrse mejor usando un número grande de sensores para rastrear cada grado de libertad en el cuerpo real. Molet et al. (1996) exige como mínimo 14 sensores para manejar biomecánicamente la postura correcta y Semwal et al. (1996) presenta un algoritmo de forma cerrada para aproximar el cuerpo que agota a los sensores. Sin embargo, muchos de los sistemas VE actuales usan sólo la cabeza y la mano y rastrean a mano. Por consiguiente, la información limitada rastreada, debe conectarse con la información del modelo humano y los generadores de diferentes movimientos en el orden a 'extrapolar' las uniones del cuerpo que no se rastrean. Esto es más que un problema de la

cinemática inversa simple, porque hay soluciones generalmente múltiples en los ángulos de unión para alcanzar la misma posición y una postura más realista debe seleccionarse. Además, las restricciones de unión deben ser consideradas para poner los ángulos de unión.

El nivel principal más bajo que se aproxima a este problema de extrapolación tiene restricciones usando cinemática inversa (Badler et al. 1993), soluciones de forma cerradas (Semwal et al. 1996) y soluciones de tabla de mirar (lookup) (Capin et al. 1995). La aproximación de la cinemática inversa se basa en un algoritmo iterativo donde un resultado final (por ejemplo la mano) coordina una escena o frame que intenta alcanzar una meta, el frame coordinado (la posición del alcance), usando un conjunto de uniones que controlan el resultado final. La ventaja de esta aproximación es que cualquier número de sensores puede ligarse a cualquier parte del cuerpo y pueden combinarse múltiples restricciones a través de la asignación de pesos. Sin embargo, esto podría bajar la simulación significativamente pues requiere cálculo excesivo. La solución de forma cerrada resuelve este problema y usa 10 sensores ligados al cuerpo, resolviendo analíticamente los ángulos de unión. El esqueleto humano es dividido en cadenas más pequeñas y cada ángulo de unión es calculado dentro de la cadena a la que pertenece. La solución de la tabla de mirar usa datos experimentales almacenados previamente.

Avatares Guiados.

Los avatares guiados son aquellos que se manejan por el usuario, pero que no corresponden directamente al movimiento del usuario. Ellos son basados en el concepto metáfora directa de tiempo real, un método que consiste de datos de entrada magnetofónicos en un dispositivo VR en tiempo real, permitiéndonos producir efectos de naturalezas diferentes pero correspondiendo a los datos de entrada. No hay ningún análisis del significado real de los datos de entrada. Para entender el concepto, nosotros podemos tomar un ejemplo de metáfora tradicional: el control del títere. Un títere puede definirse como una muñeca con miembros articulados movidos por alambres o cordones. Semejantemente los títeres del guante son muñecas cuyo cuerpo puede ponerse en la mano como un guante, los brazos y cabeza que se mueven por los dedos del operador. En ambos casos, se usan los dedos humanos para manejar el movimiento del títere.

Avatares Autónomos

Un sistema autónomo es un sistema que puede dar a sí mismo sus leyes, su conducta, como opuesto a un sistema heterónomo que se maneja del exterior. Los avatares guiados se manejan típicamente del exterior. Incluyendo avatares autónomos que interactúan con incrementos de participantes en la interacción de tiempo real con el ambiente. Por consiguiente, es probable que aumente el sentido de presencia de los participantes reales en el ambiente. Cuando estos avatares no se guían por los usuarios, ellos deben tener su propia conducta para actuar autónomamente a lograr sus tareas. Para esto se requiere construir las conductas del movimiento, así como los mecanismos apropiados para la interacción. Los avatares pueden tener una conducta, que significa que ellos deben tener una manera de dirigirse.

1.7 Aplicar Control al Movimiento.

La navegación básica involucra el uso de algún dispositivo de entrada para controlar el caminar a través del movimiento. En el contexto de NVEs, esta noción está inmensamente extendida, sobre todo cuando ellos involucran al humano como la personificación para los usuarios. La navegación involucra los problemas siguientes:

- Caminando
- La manipulación básica del objeto.
- Trazando acciones en las personificaciones.
- El apoyo general al dispositivo de entrada.
- Llevando a cabo las restricciones.

Caminando representa la navegación en su sentido básico, mientras le permite al usuario explorar el ambiente desde cualquier punto de vista

Las capacidades para la manipulación básica del objeto le permiten al usuario recoger y cambiar de sitio los objetos en la escena. Esto puede extenderse para conductas del objeto que pueden hacer que los objetos reaccionen de alguna otra manera al estar aprendiendo con la práctica

La definición de acciones en la personificación es importante por dos razones. Primero, le permite al usuario local ver lo que ellos están haciendo, por ejemplo viendo que su mano agarre un objeto. Segundo, en una sesión multiusuario esto les permite a los usuarios entender lo que los otros están haciendo intuitivamente. La definición de acciones en la personificación involucra la generación del movimiento de caminado, así como la generación de movimiento natural del brazo mientras esta manipulando los objetos

Los medios de apoyo general a dispositivos deben ser correctos para conectar cualquier dispositivo al sistema.

Las restricciones son un componente sumamente importante para cualquier navegación. Estas permiten estar de una manera precisa en el modelo de navegación. Estas se dividen en dos grupos:

- Las restricciones globales del movimiento
- Las restricciones de postura del cuerpo

Las restricciones globales del movimiento involucran un poco de conocimiento global del mundo virtual (por ejemplo la dirección y / o descubrimiento de la colisión). Un conjunto típico de restricciones para caminar, podría incluir una **restricción de inclinación** que mantiene al usuario derecho, una **restricción de colisión vertical** que los mantiene en el suelo (así como haciéndolo fácil de subir y bajar escalones o rampas) y una **restricción de colisión horizontal** que impide al usuario pasar por las paredes.

Las restricciones de postura del cuerpo mantienen la personificación del usuario en las posturas que parecen naturales, por ejemplo la cabeza no puede vagar de su posición en los hombros, el brazo puede sólo alcanzar hasta una cierta posición (figura 1-5) [Igo99].

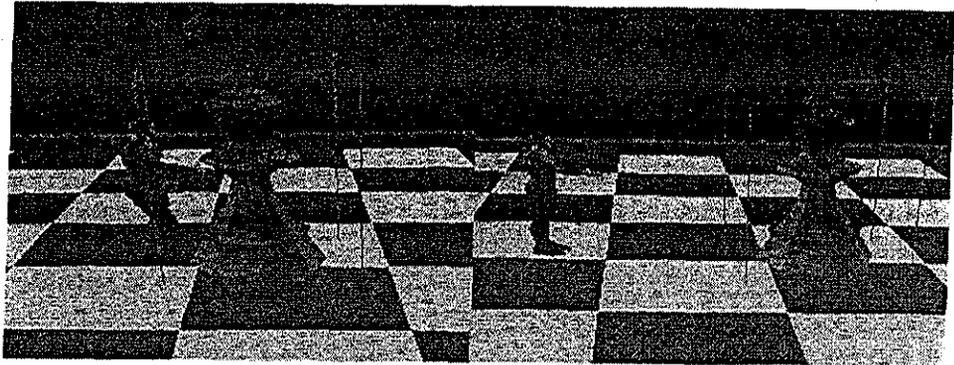


Figura 1-5 Recogiendo las restricciones introducidas por este personaje permite que los puntos de acción de los usuarios sean vistos claramente.

Aplicación a Diferentes Tipos de Avatares.

Una clasificación para el control de un avatar sería la siguiente: dirige, guías al usuario y comportamiento autónomo. Existe una mayor diferencia entre avatares controlados directamente y los otros dos tipos. Los avatares controlados directamente modifican los grados de libertad del cuerpo, sin un conocimiento extenso del método de animación (como la animación del keyframe, dinámicas, etc.). Sin embargo, los otros dos tipos de avatares (guía y autónomo) asumen un control del movimiento subyacente. Por ejemplo, las necesidades del motor de caminado de convertir el valor de velocidad, a una longitud del paso y entonces a ángulos de unión. Hay dos aproximaciones básicamente para llevar a cabo estos dos tipos de avatares: construir una biblioteca de movimientos de alto nivel, o delegar estas funcionalidades y convertir los parámetros de alto nivel a ángulos de unión.

Restricciones Globales.

La posición global de la personificación no debe inclinarse más que a un ángulo predeterminado de la escena normal. Note que estas restricciones son predefinidas y pueden ser cambiadas por el usuario si es necesario. Cuando estas restricciones son parte de la navegación y deben hacerse dentro del centro que considera toda la escena de datos VE, la máquina de navegación es responsable para realizar los cálculos. Para cada escena, la máquina de navegación realiza los cálculos siguientes:

- La restricción de inclinación global
- La restricción de elevación
- La restricción de colisión

Restricción de Inclinación Global.

Esta es principalmente útil para los participantes de escritorio y los avatares autónomos. Permite poner el cuerpo para que su normal sea inclinada, dentro de un ángulo máximo predeterminado de la escena normal (figura 1-6) [Igo99]

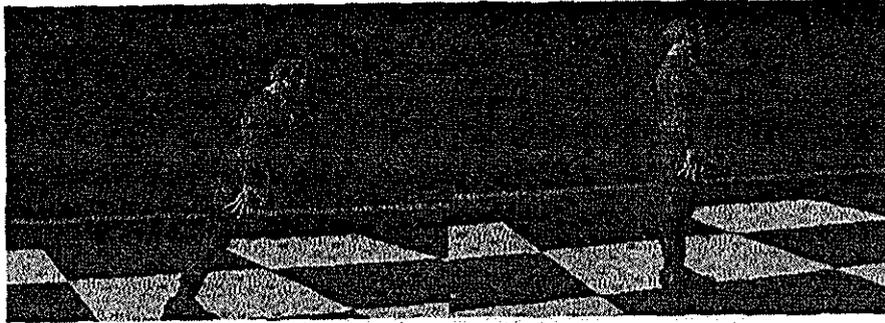


Figura 1-6. Restricción de inclinación global inclina el cuerpo normal con la escena normal.

Restricción de Elevación.

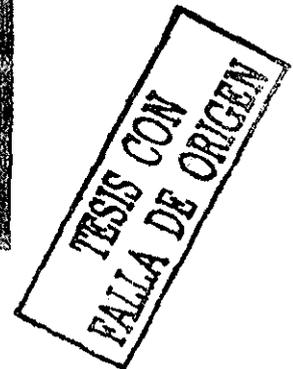
La restricción de elevación es útil para el terreno desigual, los escalones, etc. Esta se basa en disparar un rayo hacia abajo del centro del cuerpo, seleccionando las colisiones con el primer objeto y trasladando el cuerpo abajo. (Figura 1-7) [Igo99].

Restricción de la Colisión.

La restricción de la colisión previene al caminar de atravesar paredes u objetos en la escena. Esta se basa en disparar un rayo en la dirección de caminar y descubrir la intersección. La fuente del rayo es el centro de masa de la representación.



Figura 1-7. Restricción de elevación mueve el cuerpo sobre el piso.



Control Guiado para Caminar.

El movimiento de caminado, da la acción, que es considerada como la técnica de animación predefinida para la navegación. El control guiado del caminado, se logra como sigue. El

participante usa dispositivos de entrada, típicamente un ratón, pone al día la transformación de la posición del ojo del actor virtual. Este control local se usa calculando el cambio incremental en la posición del cuerpo y estimando la rotación y velocidad del centro del cuerpo. El motor de caminado usa la velocidad instantánea del movimiento, calcula la longitud y tiempo del ciclo del caminado, por el cual este calcula los ángulos de unión necesarios. El motor se basa en el modelo de caminado, guiado por el usuario interactivamente o automáticamente generando una trayectoria. Este modelo puede también incluir personificación cinemática que depende de la individualidad del usuario y se basa en la parametrización matemática que viene de los datos experimentales biomecánicos (Figura 1-8) [Igo99].

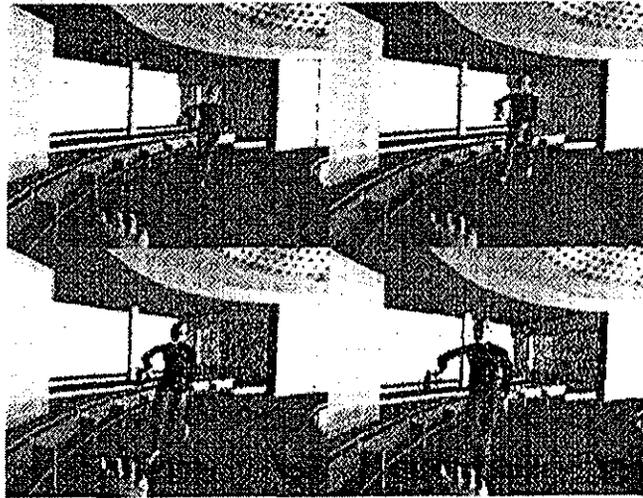


Figura 1-8. Una secuencia de caminado en tiempo real.

Control guiado para Recoger.

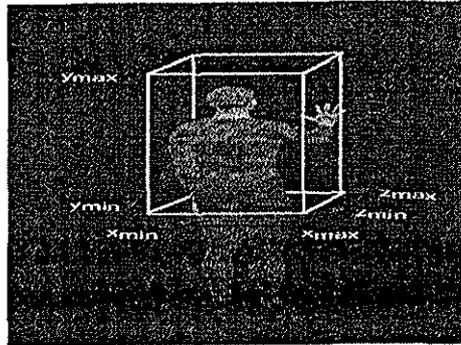
El movimiento del brazo para recoger un objeto es un problema similar al de caminar: dado 6 grados de libertad (la posición y orientación) de la mano derecha sensada con respecto al sistema coordenado del cuerpo, el motor del brazo debe calcular los ángulos de unión dentro del brazo derecho. Esto es más difícil que un problema de cinemática inversa simple, porque hay soluciones generalmente múltiples para los ángulos de unión que alcanzan la misma posición y una postura más realista debe ser seleccionada. Además, las restricciones de unión deben ser consideradas para poner los ángulos de unión.

Para la cinemática inversa en el caminado hay dos soluciones: iterativa y soluciones de mirar la tabla. El algoritmo iterativo se basa en la matriz del Jacobiano. La construcción de la matriz del Jacobiano se define bien para las estructuras articuladas y esta matriz se usa iterativamente para resolver los ángulos de unión para un conjunto dado de resultados finales. El cambio en el vector de ángulo de unión se relaciona para cambiar el dX por la fórmula

$$d\theta = J^{-1} (dX).$$

Esto nos permite a que construyamos la matriz del Jacobiano e itere hasta que el resultado final se alcance. Ver Watt(1992) para los detalles de la solución

El segundo acercamiento usa la heurística para obtener la aproximación de los ángulos de unión del brazo, dada la posición de la mano derecha. Esto se hace con la ayuda de una tabla precalculada que divide el volumen normalizado alrededor del cuerpo dentro de un número discreto de subvolumen (por ejemplo $4 \times 4 \times 4$) y almacena un trazado de subvolumen a los ángulos de unión en el brazo derecho (figura 1-9) [Igo99]. La posición actual de la mano derecha, se usa para encontrar el subvolumen en que esta reside, con respecto al cuerpo y los valores de unión son el conjunto usando este trazado (figura 1-10) [Igo99]. Las desventajas de este método son las cantidades extras de datos del subvolumen, que son más difíciles para almacenar y más difíciles de obtener experimentalmente con la discretización más fina del volumen.



TESIS CON
FALLA DE ORIGEN

Figura 1-9. Caja segura para recoger limites.

El Control Guiado para Doblar.

Usar el brazo derecho para manipular los objetos tiene limitaciones: sólo objetos dentro del alcance pueden recogerse. Para mover y bajar la mano han realizado un software de conducta de torcimiento. También el doblado usa cinemáticas inversas como en el movimiento del brazo, pero éste tiene un proceso de dos pasos. Primero, el software verifica si está dentro del alcance del brazo derecho; si esta prueba es falsa, la espina del cuerpo se ajusta usando la cinemática inversa para que el hombro esté dentro de una longitud del brazo adecuada.

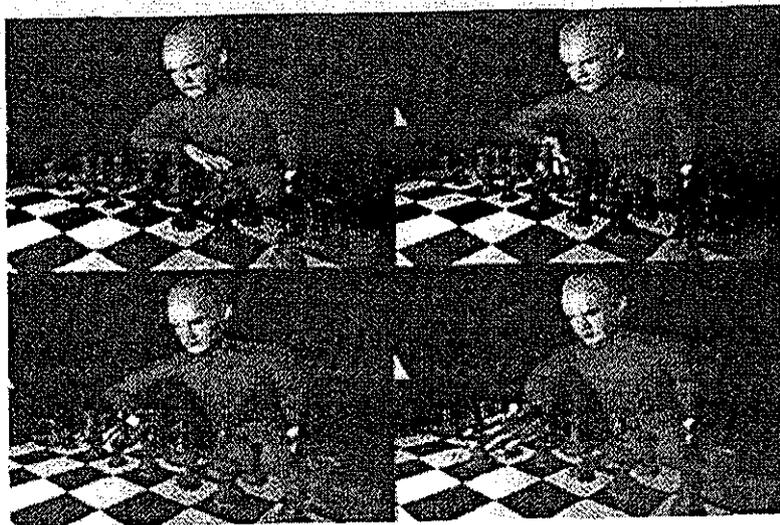


Figura 1-10. Solución de la tabla de mirar, creando una secuencia que recoge en tiempo real.

Entonces, en el segundo paso, el algoritmo de la cinemática inversa es aplicado en la cadena del brazo derecho para alcanzar el resultado final.

Control Guiado para Respirar.

Respirar es una conducta natural que da a su vida personificación, por consiguiente han llevado a cabo una conducta respiratoria. Aunque, en la vida física al respirar inflamamos y deformamos la envoltura externa del cuerpo, han encontrado una solución heurística que modifica el torso superior y los ángulos del hombro. La conducta respiratoria se basa en un aumento lineal cíclico y en la disminución de ángulos de unión particulares. El generador de conducta respiratorio toma dos argumentos: la longitud del ciclo respiratorio y la intensidad respiratoria.

Combinación de Drivers.

Más de un driver externo debe poder conectarse a la interfaz de postura humana y la tarea de la máquina es resolver los conflictos entre los drivers externos. Los drivers mandan por correo sus valores al segmento de memoria compartido asociado con ellos y la máquina de animación del cuerpo combina estos valores. Los drivers tienen el derecho de cambiar sus prioridades y pesos para cada ángulo de unión. El algoritmo para los drivers funciona como sigue: en cada escena el cliente encuentra primero al driver con la prioridad más alta entre todos los drivers. Entonces toma un peso promedio de los ángulos de unión para estos drivers. Note que la prioridad y peso de cada unión pueden cambiar sus valores a cualquier escena durante el curso de la sesión. Las soluciones generadas para resultados aceptables se muestran en (figura 1-11) [Igo99].

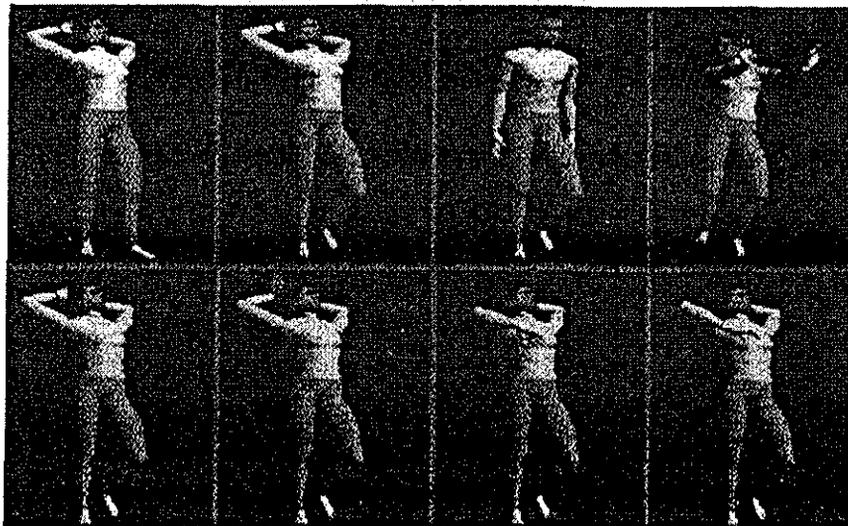


Figura 1-11. Una salida de combinación de movimiento; la parte baja del cuerpo es animada por el driver de caminado; la parte superior del cuerpo es animada por el programa de comunicación no verbal.

1.8 Actores Perceptivos Autónomos e Interactivos.

Percepción y Acción.

Los actores autónomos pueden tener conductas o una manera de dirigirse. La conducta no sólo está reaccionando al ambiente, si no también debe incluir al flujo de información, al ambiente en que actúa la persona viviente, así como la manera en que la persona codifica y usa esta información. Si considera el ambiente sintético, hecho de formas geométricas 3D, una solución a este problema es dar acceso al actor, de la posición exacta de cada objeto, en la base de datos del ambiente completo, que corresponde al mundo sintético. Esta solución podría trabajar para un mundo muy pequeño, pero no es práctica cuando el número de los objetos aumenta. Es más, esta aproximación no corresponde a la realidad dónde las personas no tienen el conocimiento sobre el ambiente completo.

El actor sintético tiene el conocimiento sobre el ambiente, localizado en una esfera centrada en él o ella. Es más, la exactitud del conocimiento sobre los objetos del ambiente, disminuye con la distancia. Esta es una aproximación más realista, alrededor de un animal o un humano hay siempre áreas dónde la sensibilidad es más importante y los estados de que si los actores pueden ver su ambiente, ellos mejorarán la planificación de su trayectoria. Esto significa que la visión es un flujo de información realista. Desgraciadamente, lo que es realista para un ser humano que camina en un corredor, parece poco realista para una computadora. Sin embargo, usando los desarrollos del hardware, es posible dar una descripción geométrica de objetos 3D, junto con el punto de vista y el punto de interés de un actor sintético, para conseguir la visión en la pantalla. Esta visión puede interpretarse entonces como la visión del actor sintético. Más generalmente para llevar a cabo la percepción, deben equiparse a los avatares con sensores virtuales.

Se define la percepción como el conocimiento de los elementos del ambiente a través de la sensación física. Para llevar a cabo la percepción, deben equiparse a los actores virtuales con los sensores visuales, táctiles y auditivos. Los sensores deben usarse como una base para llevar a cabo todos los días la conducta humana tales como la locomoción dirigida visualmente, manejando objetos y respondiendo a los sonidos y pronunciaciones. Para la audición sintética, empezamos por modelar un ambiente de sonido dónde el actor sintético puede acceder directamente la posición y la fuente de información de sonido semántico del evento audible. Simulando el sistema de sensación del tacto (haptic), corresponde aproximadamente a un proceso de descubrimiento de colisión. Pero el subsistema perceptor más importante es el sistema de visión. Una visión basada en aproximación para los avatares es un subsistema perceptor muy importante y es esencial para la navegación en los mundos virtuales. Es una aproximación ideal modelar una animación de conducta y ofrece una aproximación universal para pasar la información necesaria del ambiente al avatar, en los problemas búsqueda de caminos, anulación del obstáculo y conocimiento interior con aprendizaje y olvido.

Típicamente, el actor debe percibir a los objetos y a los otros actores en el ambiente a través de los sensores virtuales: Basado en la información percibida, el mecanismo de conducta del actor determinará las acciones que ellos realizarán. Un actor simplemente puede evolucionar en su

ambiente o ellos pueden interactuar con este ambiente o aun comunicarse con otros actores. En este último caso, se considera al actor como un actor perceptivo interactivo

Visión virtual

Usando una visión sintética nos permite saltar todos los problemas de reconocimiento del modelo y descubrimiento de distancia. En Renault et al. (1990) la visión en cada pixel de entrada tiene información semántica, que da el objeto proyectado en este pixel e información numérica que da la distancia a este objeto. Así que es fácil saber, por ejemplo, que sólo hay una mesa delante a 3 m. El actor sintético percibe su ambiente de una ventana pequeña típicamente de 30 x 30 píxeles. Cuando este puede acceder los valores del buffer z de los píxeles, el color de los píxeles y su propia posición, estos pueden localizar los objetos visibles en su ambiente 3D. Esta información es suficiente para alguna navegación local.

Se puede modelar un cierto tipo de representación virtual global, dónde el actor mantiene un nivel bajo del sistema de visión sintética, pero dónde ellos pueden acceder un poco de información importante directamente del ambiente sin extraerlo de la imagen de visión. En la visión basada en recoger, por ejemplo, un actor puede reconocer la imagen del objeto para recogerlo. Ellos pueden recibir del ambiente la posición exacta, tipo y tamaño del objeto; esto les permite caminar a la posición correcta, dónde ellos pueden empezar el procedimiento basado en los datos geométricos de la representación del objeto en el mundo. Esta mezcla de visión basada en el reconocimiento y representación de acceso global hará al actor bastante rápido para reaccionar en tiempo real. El papel de visión sintética incluso puede reducirse a una prueba de visibilidad y el reconocimiento de información semántica en la imagen puede hacerse codificando el simple color y no obscureciendo las técnicas para describir escenas. Así, pueden obtenerse la posición y la información semántica de un objeto directamente del mundo.

Audición Virtual.

En la vida real la conducta de personas o animales es a menudo muy influenciada por los sonidos. Por esta razón, se desarrolla un armazón 3D para modelar un ambiente acústico con las fuentes de sonido y micrófonos. Ahora nuestros actores virtuales pueden oír cualquier fuente de sonido (sintético o real), que debe convertirse al formato de AIFF y procesarse por el sonido dado. La descripción de la escena de sonido tiene en cuenta las restricciones de tiempos reales.

Sensaciones del tacto (Hapite)Virtuales.

Un objetivo es construir un modelo conductual, basado en una entrada sensorial táctil del ambiente, recibida al nivel de la piel. Esta información sensorial puede usarse por ejemplo en tareas tales como tocar objetos, oprimir botones u recoger objetos. Por ejemplo, en el nivel básico, los humanos deben darse cuenta de los objetos físicos, si cualquier parte del cuerpo los toca y ellos deben recoger la información sensorial, esta información sensorial es empleada en las tareas como alcanzar nuestro objeto y navegar. Por ejemplo, si un humano está parado, los pies están en contacto con el suelo. Pero durante el movimiento del caminado, cada pie experimenta la pérdida de este contacto alternadamente. Tradicionalmente estos movimientos son simulados usando dinámica y restricciones de la cinemática en la unión humana. Pero hay casos dónde la información del ambiente externo se necesita. Por ejemplo, cuando un humano baja una escalera, el movimiento debe cambiar del caminado al descenso, basado en lograr con los pasos el contacto con la escalera. Así el ambiente impone las restricciones en la locomoción humana. Proponen encapsular estas

restricciones que usan los sensores táctiles para guiar la figura humana a varias situaciones complejas además del caminado normal

Las sensaciones del tacto (haptic) simulan un proceso de descubrimiento de colisión. Para simular los eventos táctiles sensoriales, se ha diseñado un módulo para definir un conjunto de objetos sólidos y un conjunto de puntos del sensor ligado a un actor. Los puntos del sensor pueden moverse alrededor del espacio y pueden chocar con los objetos sólidos. No se descubren las colisiones si estos ocurren con objetos que no pertenecen a estos conjuntos. El único objetivo de descubrimiento de la colisión es informar al actor que hay un contacto descubierto con un objeto y qué objeto es. Las pruebas de descubrimiento de colisión normales confían en limitar las cajas o limitar las esferas para la simulación eficaz de interacciones del objeto. Un caso muy importante pero especial es el contacto entre las manos y objetos durante el proceso de recoger. La aproximación se basa en multisensores. Estos sensores son considerados como un grupo de objetos ligados a la figura articulada. Un sensor se activa con otros objetos o sensores. Aquí seleccionamos los sensores de la esfera para su eficacia en el descubrimiento de la colisión. Los multisensores en la esfera tienen ambas propiedades, longitud y táctil. Cada sensor de la esfera se encaja a su forma de la unión asociada con sus diferentes radios. Esta configuración es importante en nuestro método porque cuando un sensor se activa en un dedo, sólo las articulaciones sobre él detienen el movimiento, considerando que otros pueden moverse todavía. Haciéndolo de esta manera, finalmente todos los dedos se posicionan naturalmente alrededor del objeto. Basado en los sensores virtuales, típicamente los sensores de visión, el actor decide qué objeto recoger. El actor tiene que caminar para estar cerca del objeto. Un sistema automático decide la manera en que el actor recoge el objeto. Por ejemplo, el sistema decide usar un pellizco, cuando el objeto es demasiado pequeño para ser recogido por más de dos dedos o usar dos manos cuando el objeto es grande. La cinemática inversa se usa para encontrar la postura del brazo final. Usando multisensores, los dedos se ajustan para tener un agarre exacto (figura 1-12) [Igo99].



TESIS CON
FALLA DE ORIGEN

Figura 1-12. Agarrando diferentes objetos.

Actores Perceptivos Interactivos.

Se define a un actor sintético perceptivo interactivo (Thalmann 1996) como un actor consciente de otros actores y de las personas reales. También se asume que tal actor es autónomo. Es más, ellos se pueden comunicar interactivamente con los otros actores, cualquiera de su tipo y personas reales.

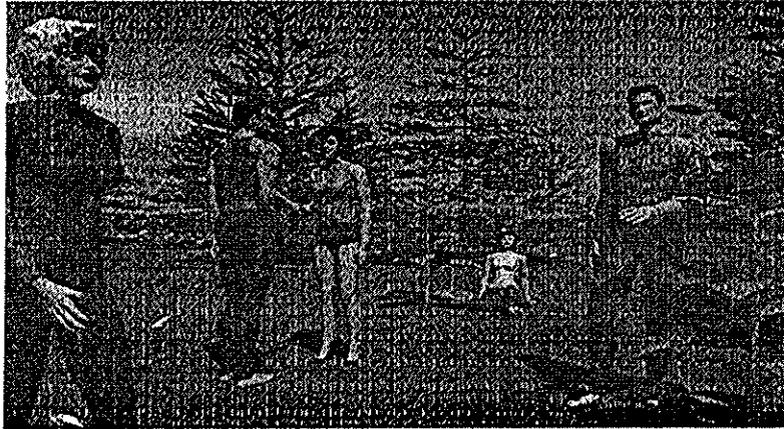


Figura 1-13. Comunicación no verbal.

Intercomunicación entre los Actores Sintéticos.

Las conductas pueden depender del estado emocional del actor. También han desarrollado un modelo de comunicación no verbal (Bécheiraz y Thalmann 1996) La credibilidad de actores virtuales se mejora por su capacidad de interpretar y usar un lenguaje no verbal. Una comunicación no verbal se preocupa por las posturas y sus indicaciones y lo que las personas están sintiendo. Las posturas son medios para comunicar y se definen por una posición específica de los ángulos del cuerpo en brazos y piernas. Normalmente, las personas no usan conscientemente la comunicación no verbal, pero ellas lo entienden instintivamente a una magnitud considerable y ellas responden a él sin ningún razonamiento explícito. Esta comunicación no verbal es esencial para manejar la interacción entre las personas (figuras 1-13 y 1-14) [Igo99]

Sensar el Mundo Real para un Actor Sintético.

Las personas reales son conscientes de las acciones de los avatares a través de las herramientas de VR, pero un mayor problema a resolver, es hacer a los avatares conscientes de la conducta de las personas reales. Los avatares deben darse cuenta que hay participantes a través de sus sensores virtuales. Un actor perceptivo tal sería independiente de cada representación de VR y de la misma manera, ellos podrían comunicarse con los participantes y con otros actores perceptivos. Pueden intercambiarse actores perceptivos y participantes fácilmente. Para la audición virtual, las restricciones de tiempos reales en VR exigen la reacción rápida a la señal del sonido y el reconocimiento rápido que esto lleva a la semántica. Acerca del sensor táctil, podemos considerar el ejemplo siguiente. El participante pone un objeto en el espacio virtual y el actor virtual autónomo intentará recogerlo y ponerlo en una mesa virtual. El actor actúa recíprocamente con el ambiente

recogiendo el objeto y moviéndolo. Como un ejemplo, hemos producido una lucha entre una persona real y un actor autónomo. Los gestos se reconocen por el sistema y la información se transmite al actor virtual, que puede reaccionar a los gestos y decidir qué actitud asumir (figura 1-15) [Igo99].

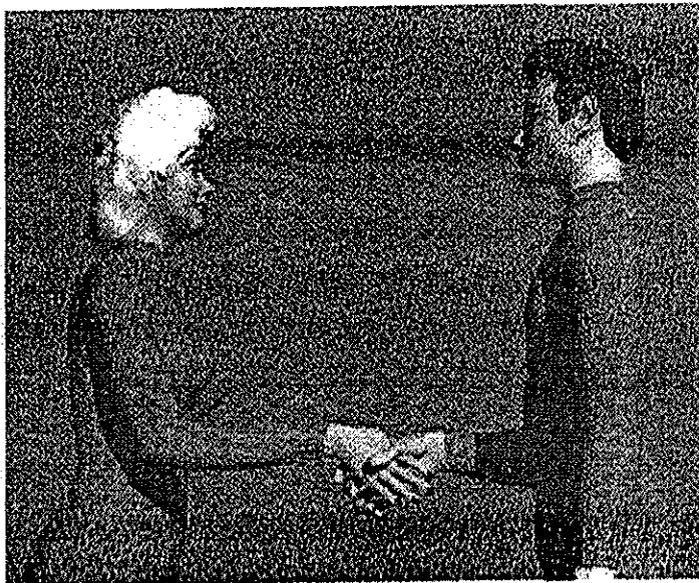


Figura 1-14. Interacción entre dos actores.

TESIS CON
FALLA DE ORIGEN

Hacia Actores “inteligentes”.

Una conducta de alto nivel usa una entrada sensorial y un conocimiento especializado. Una conducta del modelado es usar un autómatas. Cada actor tiene un estado interior que puede cambiar a cada paso del tiempo, según el autómatas actualmente activo y la entrada sensorial del actor. En lo siguiente usamos 'la conducta' y 'el autómatas' como sinónimos. Para controlar la conducta global de un actor, usamos una pila (stack) de autómatas. Al principio de la animación, el usuario proporciona una sucesión de conducta y los empuja en la pila (stack) del actor. Cuando la conducta actual acaba, el sistema de la animación hace estallar la próxima conducta de la pila(stack) y lo lleva a cabo. Este proceso se repite hasta que la pila (stack) de conducta del actor esté vacía. Algún uso de conductas de esta pila es también para alcanzar submetas empujándose con el estado actual en la pila y cambiando a la nueva conducta, permitiéndoles alcanzar la submeta. Cuando esta es una nueva conducta y continúa. Este control de conducta que usa una pila(stack) le permite a un actor ser más autónomo y crear sus propias submetas mientras ejecuta la escritura (script) original.

Introduciendo los seres virtuales aparentemente autónomos en los ambientes virtuales para cohabitar y colaborar con nosotros son un desafío continuo y una fuente de interés. Más interesante aun es la inclusión de actores autónomos en NVEs. Ellos mantienen un lugar en la reunión de personas para las situaciones geográficas y para los seres virtuales diferentes. En NVEs no vemos a nuestros corresponsales, sólo sus representaciones gráficas en el mundo virtual, lo mismo va para los seres virtuales. Por consiguiente la comunicación debe ser más natural con los seres virtuales. Éstos controlados en la computadora, donde las criaturas aparentemente autónomas habitarían los

mundos virtuales, haciéndolos más interesantes y ayudan a los usuarios a encontrar su camino o a realizar una tarea particular. Por ejemplo, una tienda virtual podría tener una tendera virtual autónoma para ayudar al cliente a encontrar cualquier cosa que ellos quieran comprar; los ambientes virtuales complejos podrían tener guías para llevar visitantes y responder a sus preguntas; en un juego, un antagonista o una referencia podría ser un actor virtual autónomo

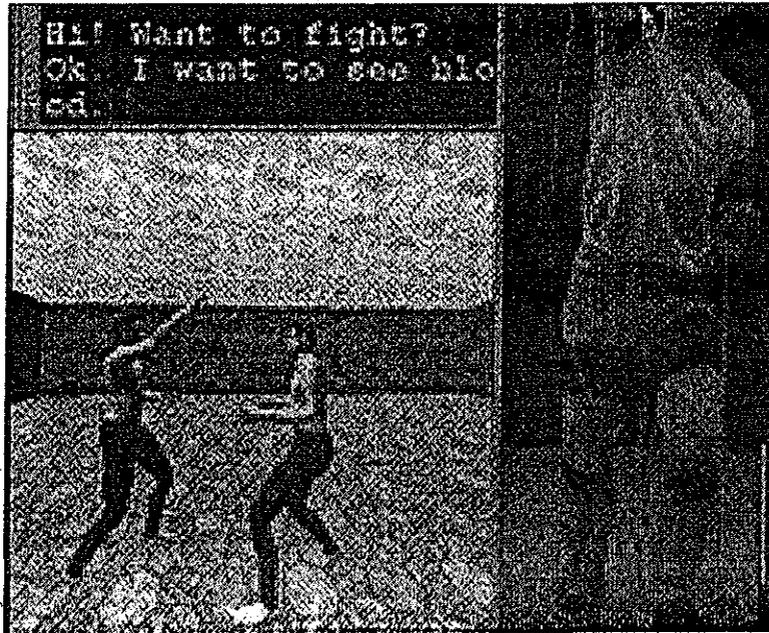


Figura 1-15. Lucha entre un participante y un actor perceptivo interactivo.

Conexión de los Avatares Autónomos con el NVE.

La conducta del avatar autónomo es típicamente afectada por VE. El programa del actor autónomo necesita recibir el estado VE del sistema de NVE. El sistema de NVE debe estar abierto, el diseñador debe tener la mínima preocupación sobre otros detalles del NVE y el programa del avatar autónomo no debe depender de una aproximación específica. La conducta debe ser fácilmente extensible, no limitada a las conductas predefinidas. Cuando los sistemas de NVE ya son muy complejos, nuestra aproximación a la conducta autónoma (AB) en NVEs está conectando los dos sistemas externamente en lugar de intentar integrarlos completamente en un solo sistema grande. Esta conexión lleva a un tipo de simbiosis entre un sistema de NVE y un sistema de AB donde el sistema de AB proporciona las inteligencias y el sistema de NVE proporciona el cuerpo para moverlo alrededor, se vea y actúe en los objetos, pero también para ver, oír y llevar la información externa al cerebro (figura 1-16) [Igo99]. Para realizar esta estrategia, el sistema de NVE debe proporcionar una interfaz externa al sistema de AB para poder engancharse. Esta interfaz debe ser tan simple que permita la conexión fácil de varias conductas autónomas. Al mismo tiempo debe satisfacer los requisitos básicos para una simbiosis exitosa de un sistema de NVE y un sistema de AB controlar su personificación y actuar en el ambiente, así como recoger la información del ambiente en que actúa.

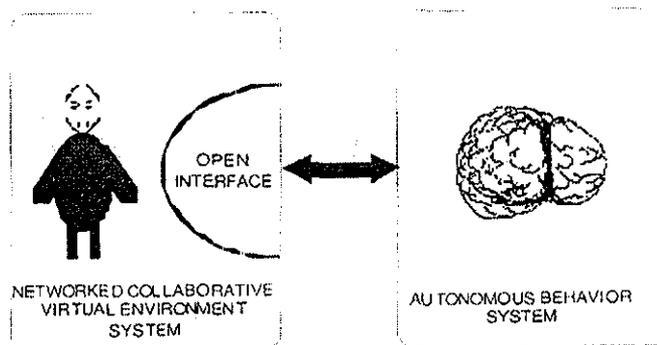


Figura 1-16 Simbiosis entre los sistemas AB y NVE.

Al estudiar las funcionalidades que el sistema de NVE debe proporcionar al sistema de AB a través de la interfaz abierta Han identificado las siguientes funcionalidades que deben mantener una simbiosis exitosa:

- La personificación
- La locomoción
- La capacidad de actuar sobre objetos
- La realimentación del ambiente
- La comunicación verbal
- La comunicación facial
- La comunicación de gestos

La personificación, o una representación gráfica, es un requisito fundamental para permitir la presencia del actor virtual en el ambiente. Aunque ésta puede ser una representación gráfica muy simple (por ejemplo un cubo texturizado), algunas de las funcionalidades más avanzadas (por ejemplo, comunicación facial con gestos) requieren una estructura más parecida a un humano como apoyo.

La locomoción es necesaria para llegar de un lugar a otro y podría involucrar la generación de movimientos del caminado o simplemente el resbalado. Esencialmente, el sistema de AB debe ser capaz de controlar la posición de la personificación en el ambiente.

La capacidad de actuar sobre objetos en el ambiente es importante porque permite al sistema de AB interactuar con el ambiente. La interfaz debe proporcionar la posibilidad por lo menos de agarrar y mover objetos

Sin alguna **realimentación del ambiente**, nuestros actores virtuales podrían ser autónomos pero ciegos y sordos. Esto es esencial, para ellos poder recoger la información del ambiente sobre objetos y otros usuarios

Si el sistema de AB es simular a los avatares, es necesario para los humanos reales ser capaces de poder comunicarse con ellos a través de un canal de comunicación más común, es decir la **comunicación verbal**. Porque la señal de audio podría en la mayoría de los casos ser insignificante al sistema de AB, el texto ASCII parece ser la manera más conveniente de unir la comunicación verbal a y desde el sistema de AB. Esto no excluye la posibilidad de reconocimiento de voz / la síntesis de la interfaz final sobre el humano, permitiéndonos realmente hablarles a los actores virtuales.

Finalmente, es deseable para un actor virtual tener **la capacidad de comunicación facial y de gestos** y por consiguiente la habilidad de tener una conducta más natural mostrando las emociones en la cara o pasando mensajes a través de los gestos.

TESIS CON
FALLA DE ORIGEN

Capítulo 2.

Generación del Modelo Humano.

Aunque el modelo humano ideal generado en computadora incluiría el músculo y tejido que se deforman durante el movimiento; así como la piel y ropa que se arrugan y estiran; el pelo que fluye, los rasgos faciales expresivos, el modelado de todos estos aspectos, así como la animación y el rendering de estos atributos son temas de investigación y el trabajo en éstas áreas todavía están en la fase experimental. De momento nuestra atención al animar aproximaciones de cuerpos reales se tiene que restringir a mejorar el movimiento sin entrar en detalle en los demás atributos. Se define como primera capa un esqueleto en donde el músculo, tejido y la piel puedan incorporarse después. Esto implica definir una estructura de esqueleto que admita movimientos de ángulos propios, dimensiones del tamaño del cuerpo apropiadas. El punto importante aquí es, modelar un esqueleto abstracto articulado. Los temas de este capítulo se basa en las referencias [Bad99], [BKK+85], [Cal91], [GM96b], [MTT90], [Zel82]

2.1 Antecedentes.

En los últimos diez años la estructura, flexibilidad y fidelidad de modelos humanos se ha incrementado dramáticamente: desde figuras de madera formadas por líneas a través de modelos de poliedros simples, a superficies curvas y aun modelos de elemento finito.

Para mejorar el realismo en movimiento se requiere de un gran número de grados de libertad en las uniones del cuerpo, pero se puede simplificar el modelo y esperar que los movimientos resultantes parezcan más naturales. Este punto de vista motiva nuestros esfuerzos en modelos de figuras humanas y animación.

Muchos sistemas de animación proporcionan medios para construir la jerarquía de transformaciones necesarias para definir un esqueleto y es bastante fácil definir una gramática simple para especificar los esqueletos. Como por ejemplo el editor interactivo para diseñar nuevos esqueletos. En los principios de los 80's advirtieron que incrementar el realismo en los modelos demandaría movimientos de control aun más complicados y precisos; por ello se debe regresar a los modelos humanos y preguntar si se debe reevaluar su estructura para tomar ventajas de mejoramientos algorítmicos. Cuando se consideró esta pregunta, se determinó que un modelo más preciso requiere del estudio de la espina dorsal y torso humanos para generar un realismo adicional en el movimiento humano

Sistemas que modelan el cuerpo humano ven el mundo a través de objetos y restricciones. Estas conectan objetos a través de relaciones geométricas. Para los objetos simples, esto es sencillo, pero no para los objetos complejos. Si un sistema representa uniones como el codo, como una restricción, ésta debe tener un factor de peso muy alto con el fin de asegurar que este nunca se separe, requiriendo pasos de tiempo muy pequeños en la simulación. Esto debe también complicar la vista del usuario de objetos tales como robots o figuras humanas, que son inherentemente articulados

2.2 Modelado Geométrico.

Los modelos geométricos actuales, se revisan brevemente con respecto a su relevancia en figuras humanas, ver ejemplo de ellos en las figuras (2.1 - 2.5) mostradas al final de este capítulo. Los modelos geométricos se clasifican dentro de dos amplias categorías: los esquemas límites y los esquemas volumétricos. Se examinan aquellas representaciones relevantes del modelado humano: en representación basada en puntos, líneas, polígonos y superficies curvas. Para la representación volumétrica, discutimos: voxeles, geometría sólida constructiva (csg), elipsoides, cilindros, esferas, supercuádricas y funciones potenciales.

2.2.1 Modelos basados en Superficies.

El modelo de superficie más simple es justamente una colección de puntos o líneas en 3D. Las superficies representadas por puntos requieren una distribución bastante densa de puntos para un modelado preciso. Nubes de puntos con profundidad se usaron hasta principios de los 80's para modelos humanos, tomando ventaja en la rapidez de despliegado y en sus transformaciones jerárquicas estas producen el efecto de profundidad perceptual activado por el movimiento de puntos. Una técnica relacionada para obtener rapidez de despliegado es el uso de anillos en paralelo. Esta técnica es usada en uno de los primeros modelos de figuras humanas y subsecuentemente en un vídeo de música, de Mick Jagger, "Hard Women" de producciones Digital.

Polígonos.

Los modelos poligonales (poliedros) son unas de las representaciones mas comúnmente encontradas en aplicaciones gráficas. Los modelos son definidos en redes de polígonos formando un poliedro 3D. Cada polígono (primitivo) consiste de una estructura conectada por un vértice, una arista y una cara. Los polígonos tienen tamaño, forma y posición, ellos llenan completamente la superficie requerida en alguna resolución. Los modelos poligonales son relativamente sencillos para definir, manipular y desplegar. Ellos son los modelos más comunes procesados por hardware y software en estaciones de trabajo. En general, los polígonos son los mejores para modelar objetos, refiriéndose a estos por tener una superficie plana, aunque se requiere un número bastante grande de polígonos para representar objetos complejos, esto significa cientos de miles de polígonos.

Todos los modelos viables de figuras humanas interactivas son hechos con polígonos, primeramente porque el polígono es la primitiva más fácil de manejar por una estación de trabajo moderna tal como la máquina de Silicon Graphics o por las máquinas actuales basadas en Pentium y con tarjetas gráficas especiales. Con el sombreado suave en tiempo real, los modelos de polígonos de complejidad moderada (varios cientos de polígonos) pueden verse aceptablemente como humanos; modelos de piel precisos requieren miles de polígonos. Una cara real solo debe requerir de 2 a 3 mil polígonos, si esto tiene que ser animado y si los polígonos son la única fuente de detalle.

Los modelos de polígonos son también numerosos por citar algunos que tienen un nivel interesante de detalle y que han sido usados en el software de maniquí de Biomechanics Corporation of América, así como también en varias películas tales como "Tony de Peltrie" de la Universidad de Montreal y modelos de actor sintético detallado de Marilyn Monroe y Humphrey Bogart de Daniel Thalmann y Nadia Magnenat - Thalmann. Ellos aun modelan detalles de piel deformada aplicando fuerzas físicas y mallas poligonales. Los modelos normales tienen unos pocos cientos de polígonos. Los Modelos más precisos obtenidos de búsquedas actuales de cuerpos reales tienen varios miles de polígonos.

Superficies Curvas.

Aunque los polígonos son buenos para representar superficies planas, esfuerzos considerables han sido expandidos, determinando formulas matemáticas para superficies curvas. Muchos modelos de objetos de superficies curvas, son formados por una o más funciones paramétricas de dos variables. Cada superficie de curva es llamada un "patch"; los "patches" deben ser unidos a lo largo de sus fronteras, para formar superficies más complejas. Los "patches" usualmente son definidos bajo el orden del polinomio (típicamente cúbicos) dando al "patch" las propiedades matemáticas fácilmente calculadas tales como normales y tangentes de superficies bien definidas y capaz de calcular las condiciones de continuidad en "patches" de aristas adyacentes. La forma de un "patch" se deriva de los puntos de control o vectores tangentes; hay tipos de interpolación y aproximación. Hay numerosas formulaciones de superficies curvas, incluyendo: Bezier, Hermite, bi cubica, B spline, Beta spline y polinomios racionales.

Varios modelos de figuras humanas han sido contruidos de "patches" Ellos son excelentes para animación. Proveen ese cuidado suficiente que es tomado para modelar ángulos de conexiones. Modelos de superficies curvas fueron usados en piezas gimnásticas y el premio al ganador de la academia "Tin Toy".

Recientemente, Kim et al. [MK95] desarrolló las curvas de quaternion de forma cerradas en S^3 que usa Bezier, Hermite, B-spline (o de hecho, cualquiera) mezclando las funciones y pudo calcular las derivadas paramétricas de alto orden sobre las curvas. Éste tiene grandes noticias para aplicaciones que deben calcular y deben perfeccionar o deben integrar a lo largo de las curvas de orientación fijas, pero no ayudan grandemente en el control diferencial u optimización sobre la propia forma de la curva, desde entonces no provee un método simple correspondientemente para diferenciar la curva con respecto a los puntos de control del quaternion. No obstante, la habilidad de especificar la forma cerrada de las curvas de Hermite sobre S^3 por keys quaternion y velocidades angulares en las keys parecen prometer para el uso en los sistemas de animación de keyframe, los métodos convenientes dados por visualizar las curvas del quaternion.

2.2.2 Modelos de Volumen y CSG.

Los modelos 3D son compuestos de elementos sin intersección dentro de una partición especial, tal como voxeles o octrees, o bien creados por combinaciones de volúmenes inherentemente primitivos 3D.

Modelos Voxel.

El primer modelo volumétrico que examinamos es el modelo voxel. Aquí el espacio es completamente llenado por un mosaico de cubos o paralepípedos llamados voxels (elementos de volumen). Usualmente algunos esquemas de optimización de almacenamiento son requeridos para trabajar en detalle (espacios de $1k \times 1k \times 1k$). Técnicas especiales son necesarias para calcular superficies normales y sombreado para suprimir las cajas de renglones de voxel primitivos. Los voxeles son obtenidos comúnmente en dominios médicos, ésta es altamente considerada para fines de diagnóstico.

Los modelos voxel son las bases para muchos trabajos de visualización científica en imágenes biomédicas. El detalle posible para modelos humanos es limitado solamente por la resolución del sensor. Precisar la forma del ángulo del hueso debe ser visualizado, también como los detalles de características fisiológicas internas y externas. Desplegados en tiempo real de interacciones y visualizar las imágenes basadas en voxeles son difíciles, requiriendo un conjunto de imágenes de baja resolución o hardware especial.

Geometría Sólida Constructiva (CSG).

Una de las técnicas de modelado más eficiente es la geometría sólida constructiva (CSG). Los objetos primitivos no son limitados a cubos (uniformes), allí hay un número simple de primitivas tales como cubos, esferas, cilindros, etc. Cada primitiva, es transformada o deformada y posicionada en espacio. Combinaciones de primitivas o de objetos combinados previamente se crean por las operaciones booleanas. Un objeto además existe como una estructura de árbol que es evaluada durante el rendering.

La CSG ha sido usada con gran ventaja en partes mecánicas de modelado, pero no ha sido seriamente usado para modelado de cuerpos humanos. Al lado del parecido mecánico creado, desplegados de tiempo real no son posibles a menos que los primitivos CSG sean poligonizados dentro de superficies. Cuando el conjunto de primitivas es restringido de un camino a otro, puedan construirse algunos modelos humanos útiles.

Aunque en visión por computadora sí se han empleado los modelos de cilindros generalizados para representar la forma del cuerpo humano, para segmentación y seguimiento en secuencias de imágenes ([Mar85],[Hog84],[Roh94])

Sistemas Primitivos Únicos.

La generalidad del método de geometría sólida constructiva – con su multiplicidad de objetos primitivos y métodos caros de desplegado de ray tracing lento – son reducidos frecuentemente para ganar eficiencia en la construcción del modelo, evitar combinaciones booleanas mas que unión y incrementar la rapidez de desplegado. La idea es restringir primitivas a un tipo, entonces diseñando la manipulación y desplegado de algoritmos, toma ventaja de la uniformidad de la representación. Voxels podrían ser considerados como un caso especial, donde las primitivas son todas de ejes coordenados alineados y cubos posicionados íntegramente. Otros esquemas son posibles, por ejemplo, usando elipses, cilindros, Supercuádricas o esferas.

Las Elipsoides han sido usadas para modelar figuras sencillas. Ellas son buenas para alargamiento, para objetos simétricos redondeados.

Los Cilindros han sido usados para extensión de modelos, de objetos simétricos. Los Cilindros elípticos fueron usados en un sistema primario de modelado humano. Estos primitivos sufren de problemas de conexión de ángulos y de representaciones pobres en secciones cruzadas de segmentos del cuerpo.

Las Supercuádricas son una generalización matemática de esferas, que incluyen una interesante clase de formas, dentro de un marco de trabajo único: esferas, elipsoides, y objetos que parecen arbitrariamente cerrados como prismas, cilindros, y estrellas. Los

parámetros simples controlan la forma de estas deformaciones a través de miembros de las clases simples y naturales. Las Supercuádricas son primeramente usados para modelar objetos hechos por el hombre, pero cuando son traslapados pueden dar la apariencia de caras y figuras

Con visión por computadora las supercuádricas han sido usadas para segmentación y seguimiento ([Kak97])

Las esferas tienen una simplicidad geométrica, que rivaliza a la de puntos simples: justamente añadiendo un radio Normalmente las esferas son dibujadas como objetos regulares 3D En un modelado humano, este camino tiende a mirarse como una gran molécula desigual. Alternativamente, las esferas deben ser tratadas como “escalas”, sobre el objeto modelado; en este caso una esfera render es una esfera de disco plano Con las esferas de traslape, el resultado es un sólido de sombra suave que modela volúmenes de curvas bastante bien. Una figura humana natural fue hecha, en un sistema primario TEMPUS, este método no puede controlar adecuadamente el traslape esfera/disco durante la animación.

Funciones Potenciales.

Una generalización interesante de esferas, que resuelven algunos problemas principales de modelado, es considerar el volumen, como una función potencial, con un centro y un campo de función (función polinomial o exponencial) que decrece monótonicamente del centro externo. El tamaño o superficie de la función potencial se determina activando un valor de umbral para el campo. Lo que hace ésto más interesante, son las funciones potenciales que actúan como fuentes de energía Las funciones potenciales fueron usadas originalmente en modelos de moléculas, entonces los átomos exhiben exactamente esta forma de campos de conducta, pero los modelos tiene un “parecido” natural sorprendente y han sido usados para grandes efectos en formas orgánicas de modelado incluyendo figuras humanas y animales Las desventajas principales para funciones potenciales, caen propiamente generando funciones de traslape numerosas y desplegadas muy lentos Estas permanecen como una posibilidad interesante para modelos altamente realísticos.

2.3 Representando Figuras Articuladas.

Un modelo de cuerpo humano, bajo su piel tiene una representación de un esqueleto Esta representación del esqueleto sirve para definir las partes del movimiento de la figura. Aunque esta representación es posible para modelar cada uno de los huesos en el cuerpo humano y mover relativamente uno a otro, para muchos tipos de análisis geométricos esto es suficiente para modelar los segmentos del cuerpo en términos de sus longitudes y dimensiones y los ángulos en términos de rotaciones simples. Hay algunos grupos de ángulos más complejos tales como el hombro y espina donde dependencias inherentes a través de varios ángulos requieren más cuidado y un modelado sofisticado.

El interés incrementado en años recientes en sistemas orientados a objetos, es grande debido a la realización del diseño de un sistema, este debe comenzar con un entendimiento profundo de los objetos que manipula. Esto parece particularmente verdadero en un sistema de modelado geométrico, donde la palabra "objeto" toma muchas de sus connotaciones menos abstractas. Esto ha sido durante un tiempo un proverbio en la comunidad de software de interfaz de usuario que un sistema de estructura básica pobremente diseñada no puede ser reparada mejorando la interfaz y además que un sistema bien diseñado imparte fácilmente así mismo a una interfaz elegante.

La representación de figuras articuladas compuestas de segmentos conectados por ángulos se diseña con varios criterios en mente:

- Este debería ser de propósito general. Debería ser capaz de representar muchos tipos de figuras de topología de árbol estructurada. No ser de difícil codificación para representar un tipo específico de figura, tal como una figura humana o un manipulador de robot particular.
- Tendría una notación bien desarrollada de articulación. Además concentrado sobre representaciones para formas geométricas primitivas, tales formas pueden ser conectadas juntas y estas se adecuan relativamente una a otra.
- Debería representar objetos de árbol estructurado a través de una jerarquía. El algoritmo de posicionamiento de cinemática inversa, puede calcular y mantener la información necesaria para simular ciclos cerrados.
- Debería ser fácil de usar. La vista del usuario externo de las figuras debería ser lógica, clara y fácilmente entendible. Entendiendo que las figuras no deberían requerir ningún conocimiento de la aplicación interna y no deberían requerir ningún conocimiento avanzado de robótica o mecánica.

2.4 Modelo del Esqueleto Humano.

2.4.1 Representación del Esqueleto.

Un esqueleto puede representarse, por una colección de objetos rígidos simples, conectados juntos con los ángulos. Los ángulos son normalmente rotatorios, pero también pueden estar resbalando (o prismático). Cada ángulo rotatorio puede permitir la rotación en una, dos, o tres direcciones del ortogonal; éstos son los grados de libertad (DOF) del ángulo. Una aproximación detallada del esqueleto humano puede tener tantos como 200 grados de libertad, aunque frecuentemente menos le basta. Las restricciones en el rango aceptable de

movimientos, para un ángulo pueden ser aproximadas, limitando el ángulo de la rotación, en cada una de las direcciones de rotación (anexo A código de restricciones). Los objetos individuales que comprenden el esqueleto, son cada uno definido en sus propios sistemas coordenados locales y se congrega en una figura reconocible en un sistema coordenado global por una serie anidada de transformaciones. Un miembro articulado simple, se construye aplicando rotaciones y translaciones locales a los bloques definidos en sus propios sistemas coordenados locales. Un segmento del esqueleto es una parte del cuerpo formada por esferas, cilindros y transformaciones (translaciones y rotaciones) que ayudan a formar ya sea el brazo izquierdo, o el brazo derecho, la pierna izquierda, o la pierna derecha, los hombros, la espalda, etcétera. Y todo el conjunto de segmentos crea el modelo del esqueleto humano.

2.4.2 Jerarquías.

Al construir un esqueleto de segmentos, creará una docena o más de segmentos. Para poner todos los segmentos juntos y estos se muevan como una unidad, se usa una pila (stack). Si quiere mover la mano con el brazo y mover las piernas con las caderas, se ponen en la pila los segmentos del esqueleto juntos usando una jerarquía. Una jerarquía es una manera de decir cómo el esqueleto se conecta. La jerarquía dice como el pie se conecta al tobillo, el tobillo se conecta a la espinilla y así sucesivamente. Técnicamente, la jerarquía se parece a un pedazo de un árbol, con cada conexión que forma una rama. El tronco es el padre de las ramas y las ramas son padres de las ramitas. Si mueve un segmento del hijo, diga una ramita, no moverá al padre (rama o tronco). Sin embargo, si mueve el tronco, todos los hijos, las ramas y las ramitas, se mueven.

Simplemente como un árbol, una jerarquía debe tener un tronco, un solo padre que controla todas las otras ramas. En un esqueleto humano, el tronco son normalmente las caderas o la pelvis. La jerarquía básica de un cuerpo humano. (Figura 2.0) [GM96a,b]. Las caderas son el padre de todas las uniones. Si mueve las caderas, las demás partes también se mueven.

La elección de la raíz de la figura, es particularmente importante para el algoritmo de la cinemática inversa, entonces el algoritmo opera sobre cadenas de segmentos dentro de la figura. Al menos un punto sobre la figura debe permanecer fijo en el espacio. Porque la representación interna de la jerarquía se separa, el usuario mantiene un observador consistente de la transformación a través de una unión, a pesar de cómo la figura sea en su raíz.

Estas transformaciones anidadas en la jerarquía, aseguran que los segmentos, heredan las rotaciones, aplicadas a los ángulos superiores, en el árbol; por ejemplo: una rotación aplicada, digamos en el ángulo del hombro, causa que el brazo entero gire y no sólo el segmento del brazo superior. Un ángulo del esqueleto necesita ser especificado como la raíz del árbol; las transformaciones aplicadas a éste ángulo mueven el esqueleto entero en el sistema coordenado global. La opción de que el ángulo, tiene que servir, como

la raíz no es pertinente y es conveniente poder reestructurar una jerarquía existente cuando quiera un nuevo ángulo alrededor de la raíz. Sin tener en cuenta cómo se crea, una definición del esqueleto se requiere que se especifique mínimamente las longitudes individuales del segmento del cuerpo, los grados de libertad del ángulo y la jerarquía global de la estructura. Los esqueletos más complejos pueden construirse colocando los segmentos en una jerarquía de árbol estructurada

2.5 Crear el Esqueleto Humano (Modelo).

Para poder crear el esqueleto es necesario que se cuente con una estructura de datos (anexo C) que contenga las variables que simulan un esqueleto. El esqueleto es simulado por medio de esferas y cilindros conectados por uniones y guardados en segmentos aparentando un esqueleto humano. El código para crear el esqueleto es de segmentos de sólo esferas y cilindros, aunque simple, le muestra el modelo de un cuerpo humano. En la dirección <http://ligwww.epfl.ch/~plaenker/BodyFitting/> se encuentran varias figuras humanas tanto de hombre como de mujer, este considero es un buen ejemplo con una rica variedad de modelos humanos.

2.5.1. Restricciones

Los mecanismos de estructuras de ciclo cerrado, son manejados a través restricciones satisfechas. Las restricciones en el rango aceptable de movimientos para una unión pueden ser aproximadas limitando el ángulo de rotación en cada una de las direcciones de rotación a cada unión, el código se encuentra en el anexo C.

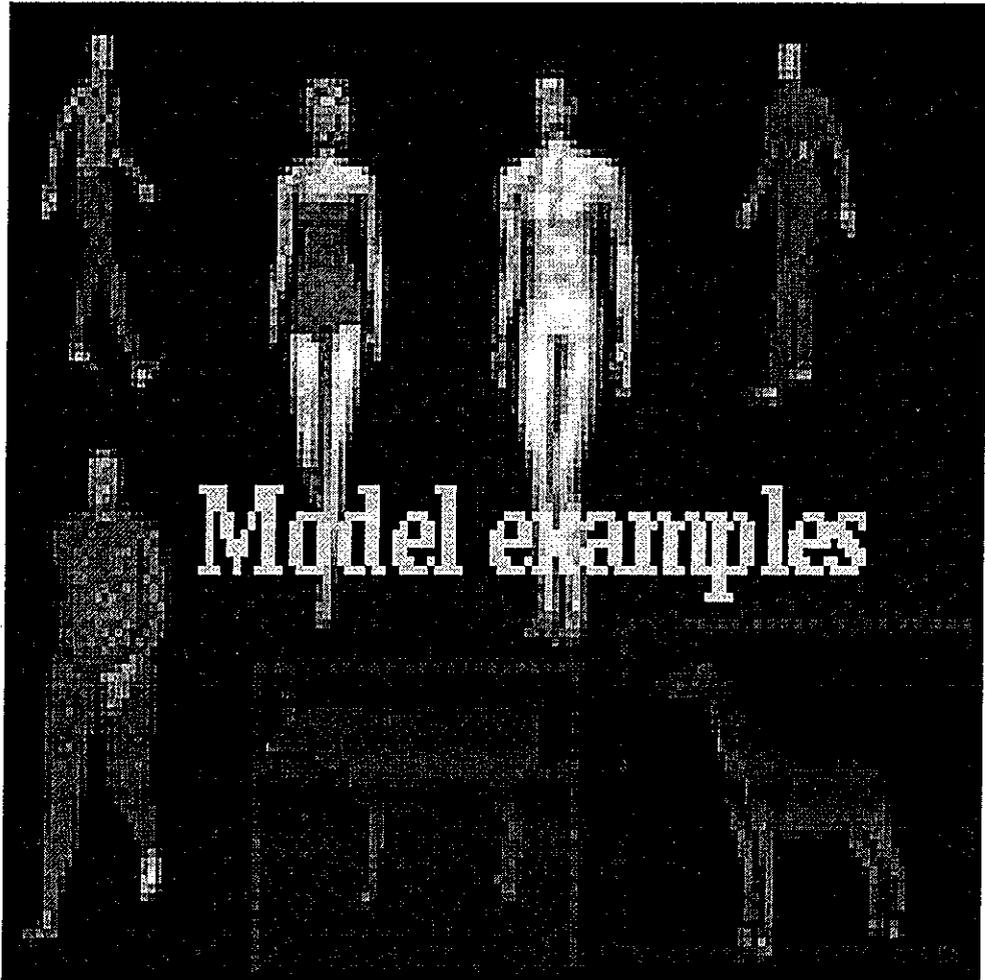
2.5.2. Condiciones Iniciales.

La figura humana puede ser abstraída como un objeto dentro de una cierta pose o configuración para cualquier especificación de todos los ángulos de unión. Mientras cualquier pose pueda ser representada por un conjunto de ángulos de unión, Esto no siempre es posible para suministrar un conjunto de ángulos razonables y totales. En este modelo se presenta, una configuración dando el conjunto de ángulos de unión, para representar la visualización del modelo, que genera la pose, cuyo código esta en el anexo C.



Figura 2.0. Jerarquía Básica de un Cuerpo Humano.

TESIS CON
FALLA DE ORIGEN



Modelsz. No.10.

Figura 2.1. Ejemplos de Modelos Geométricos.

Nó.	Dirección de la World Wide Web	Ligas 3D / Modelos
1	www.softdevspb.com/plugins/AniLabPage/links.htm	www.3dcafe.com , www.meta3d.com www.3dtotal.com , www.3dshop.com www.3dmodels.ru/models/
2	www.hash.com/users/alweb/April/newmay.html	Indian1 (modelo)
3	www.dickblick.com/categories/anatomicalmodels	21603-Obww (modelo de madera)
4	www.jungle.or.jp/uribo/Amapi-Poser/MakingPoserMT-E.html	Zoomup, Magentcircle, Morphlt, MagentMorphlt, final. (modelos)
5	Http://staigetland.com/amapi/sweater/index.html	Im1, im2, im3, im4 (modelos)
6	Http://3drender.com/nurbana	Wommod, mouthcenterb (nurbs elemento finito) (modelos)
7	Http://ece.uwaterloo.ca/~h-anim/ Humanoides	Sm_bimos_jc (modelo de puntos) Sm_bimos_001 (modelo esferas) Sm_hiro_001 (modelo hombre) Sm_yt_001 (modelo mujer) Sm_egghead_001 (modelo combinado) Makinglegghead001 (modelo)
8	www.comet-cartoons.com/toons/3ddocs/bodymodeling/index.html	Step15 (modelo mujer)
9	www.jupiterstudio.com	Sunset (modelo)
10	www.jupiterstudio.com/femalebodytutorialFrame.html	Modelsz, layedmodel (modelos)
11	Http://ligwww.epfl.ch/~babski/StandardBody	Sphereman (modelo)
12	www.dlc.fi/%7Eops/tutorial/human.htm	Me1, manfin, manfins (modelos) Makingprofiles (refmini), Modingleg (leg1, segs, extleg, edfront, edside), Torso (ed2front), Arm (ed2side, armface, halfbody, halfshad) Hand (fingref, fingtop, fingside, finged2, finged, refnail, nailface, nailext2, finfing, thumb) Conectar dedos a la palma (fingers, palm palmtop, palmiside, pinhand, badcon, remface, weld, goodcon)
13	www.3dspline.com/modeling	Shoe, shrefsf, shoefr, shoesid, shoefin, shoesth, connections, manhalf, meshsmooth, manfin, manfins
14	www.dlc.fi/~ops/tutorial/clothing/clothing.htm	Marine (modelo)
15	Http://moon.ouhsc.edu/dthompsn/namics/mass.htm	Segments (modelo)
16	Http://maxrovat.sns.hu/subdiv/	Boxman, edge loop, box man2 (modelos)
17	www.nlm.nih.gov/research/visible/vhp_cnf/mueller/paper.htm	Page1185 (modelos)
18	www.3dcafestore.com/powermodels.html	Pmpeople (modelo)
19	Http://store1.yimg.com/l/3dcafe_1661_2774616	Completo man (modelo)
20	www.mannequins.nl/mannequins-uk.htm	Mov he front-k, vc-wig-k (modelo)
21	www.animationpitstop.com/models/users/characters1.shtml	Maqmade, women3 (modelo)
22	Htp://amber.aae.uiuc.edu/~afb/swift1lgeo.html	Swift1lgeo_v2 (modelo)
23	www.cg.tuwien.ac.at/studentwork/CESCG-2000/Rfil	Korn/ image14

Figura 2.1. Tabla de Direcciones de los Modelos Geométricos.

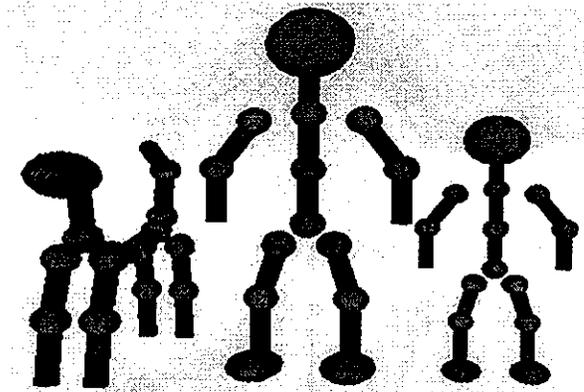
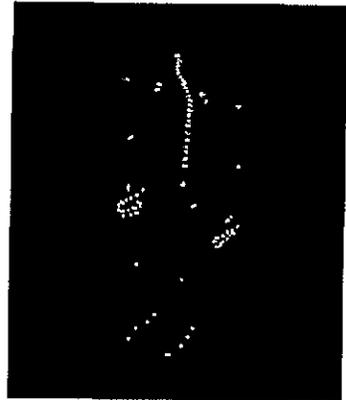
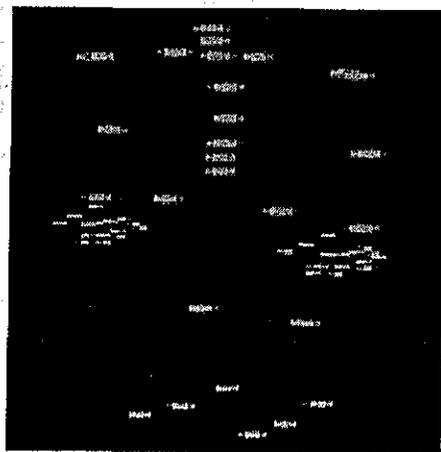


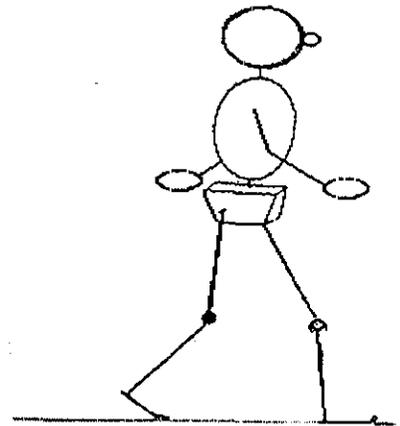
Image14.No.23.



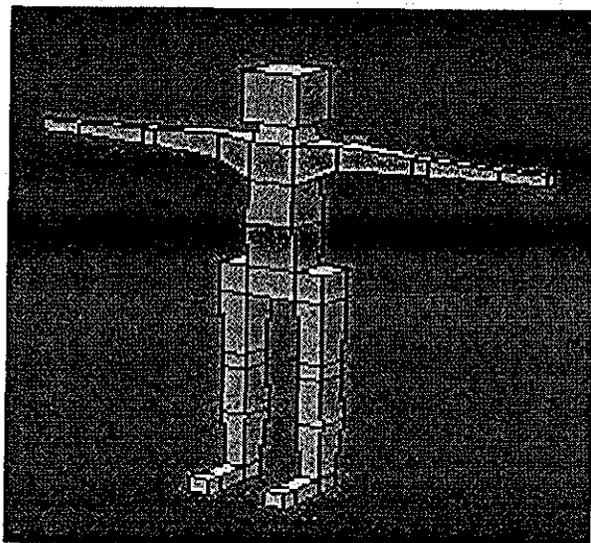
sm_bims_jc.No.7.



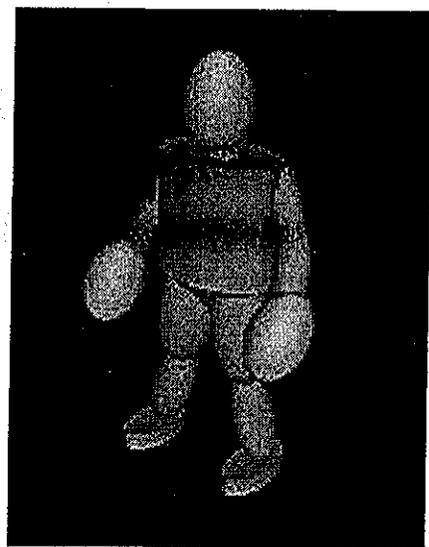
Sphereman.No.11.



segments.No.15.

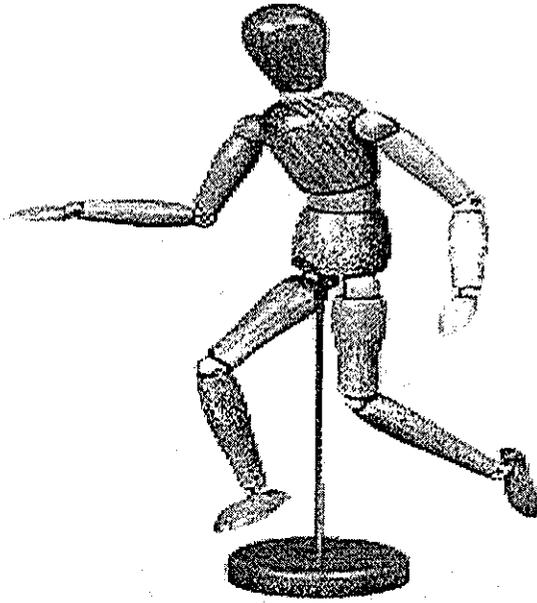


Boxman.No.16.



sm_egghead_001.No.7.

Figura 2.2 Modelos basados en Superficies.



216030-Obww.No. 3.



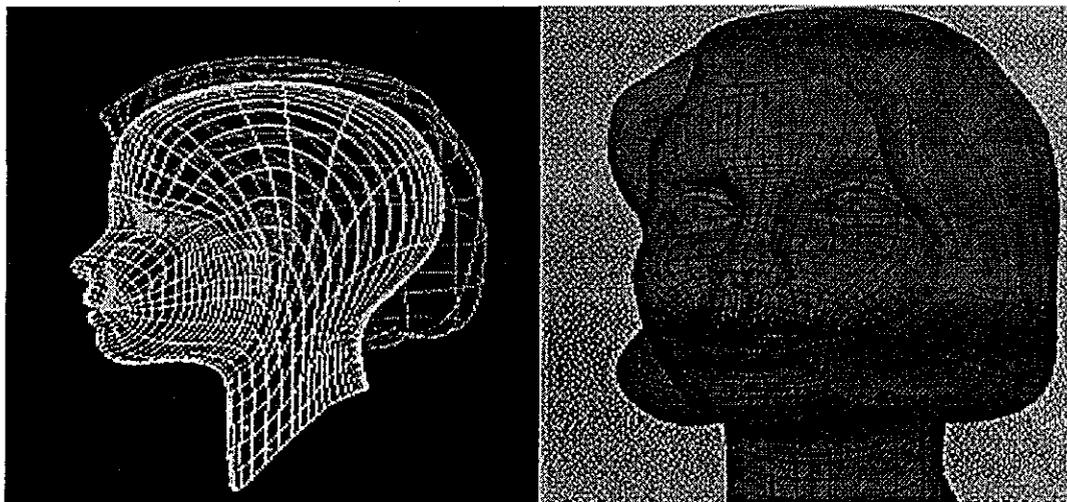
sunset.No. 9.

Figura 2.3 Modelos Geométricos.

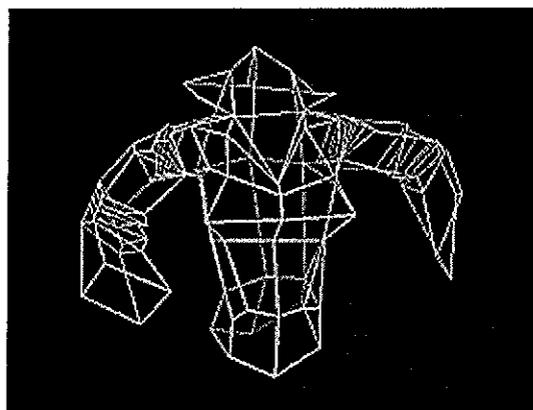
TESIS CON
FALLA DE ORIGEN



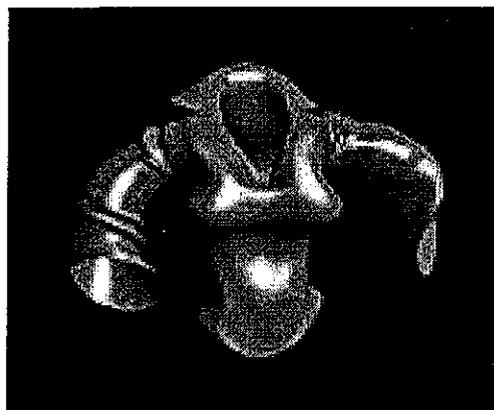
Wommod.. No. 6.



Mouthcenterb.N0.6

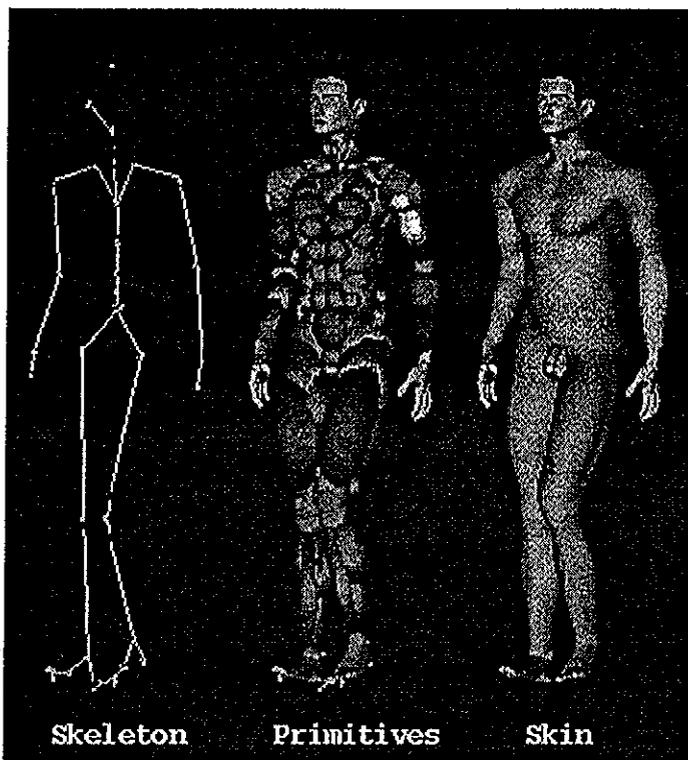


im1.No.5.

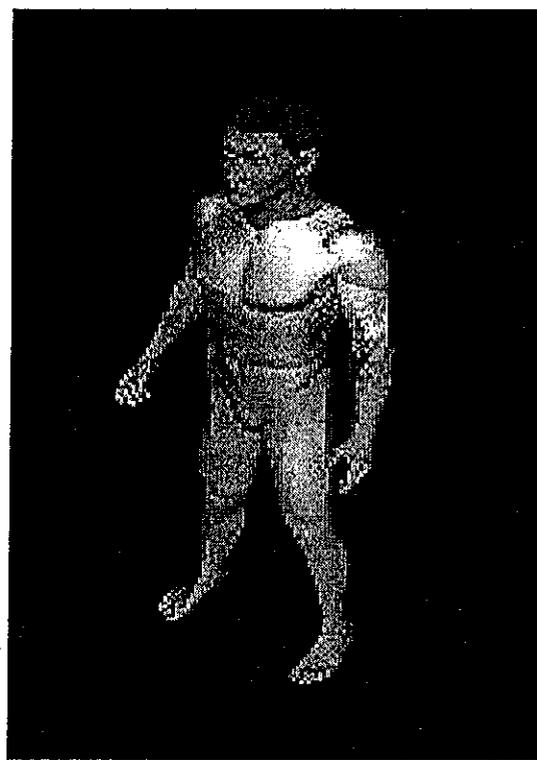


im4.No.6.

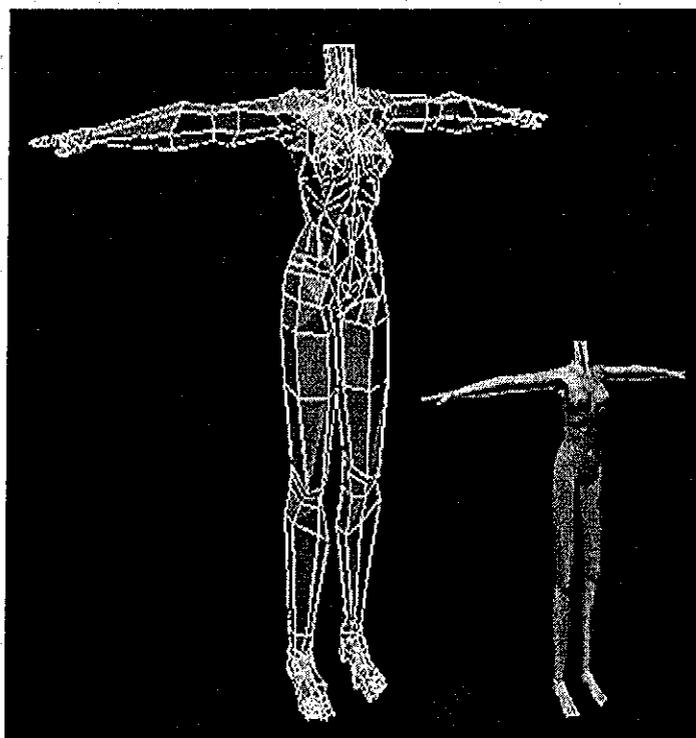
Figura 2.4. Modelos Geométricos.



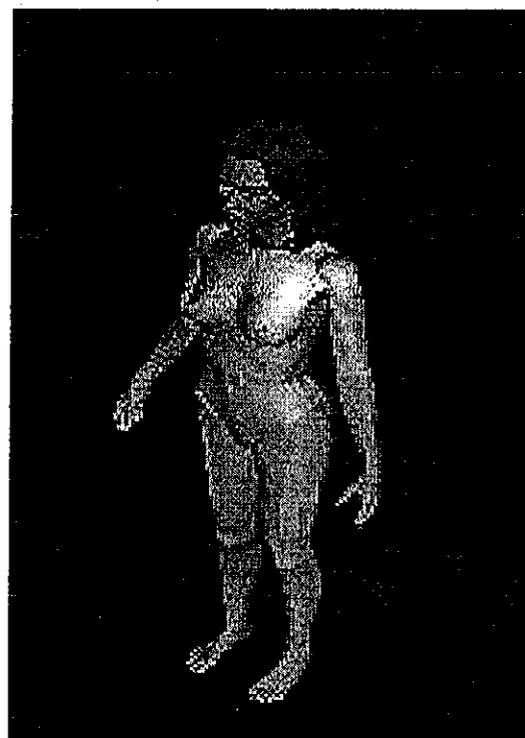
Layelmodel.No.10.



sm_hiro_001.No.7.



Step15w.No.8.



sm_yt_001.No.7.

Figura 2.5. Modelos Geométricos.

Capítulo 3.

Análisis del Caminar Humano.

Una figura humana desafía la universalidad y complejidad que tiene el movimiento humano. Las personas son al menos el objeto de interés más común y la estructura más compleja. El interés en representar movimientos parecidos al humano, tiende naturalmente hacia las estructuras de articulación más realistas. En general, diseñar movimientos de modelos altamente articulados. Puesto que la precisión del movimiento humano es difícil de sintetizar, capturar el movimiento es una alternativa popular, pero uno debe de reconocer su limitada adaptabilidad y su dominio específico. Entonces para encontrar los elementos principales que intervienen en el movimiento del caminado humano y explicar su proceso, es necesario estudiar el comportamiento del movimiento humano, analizar su movimiento y con esto describir las partes del caminado humano, los resultados se muestran en este capítulo. El Tema de este capítulo se basa en las referencias [Bad75], [Bad99], [BS79], [BWKE91], [GM96b], [RP2002]

3.1 Antecedentes.

Algunas aplicaciones del movimiento de figuras humanas en 3D son: Simulación de caída, análisis de movimiento, estimación del lugar de trabajo, notación de movimiento o baile, entretenimiento, entendiendo el movimiento, agentes en ambientes virtuales

1. Simulación de caída Esta aplicación se caracteriza por el posicionamiento no interactivo y fuerzas basadas en cálculos de movimiento con análisis de impactos de fuerza para afectar regiones del cuerpo

2. Análisis de movimiento Los Atletas, los pacientes con discapacidades psicomotoras, los actores, o los animales que fueron fotografiados en movimiento por una o más cámaras fijas calibradas. La información fue correlacionada entre observadores y reconstruida como

puntos de datos en 3D. Estos datos podrían ser filtrados y diferenciados para calcular velocidades, aceleraciones, fuerzas y torques. La visualización de los datos originales validaron, el proceso del conjunto de datos, pero requirieron modelos de figuras humanas. Frecuentemente los modelos de figuras, sirven como un rol de apoyo para mejorar el desempeño atlético, en análisis biomecánico, en movimiento de caricaturas y en terapia de entrenamiento o físico [Win90].

3. Estimación del lugar de trabajo. El dominio de este problema se caracterizó por requerimientos de posicionamiento interactivo del cuerpo y el análisis basado sobre inspección visual de modelos 3D. La interacción rápida con el desplegado del modelo, provee alimentación dinámica al animador en el lugar de trabajo

4. Notación de movimiento o baile. El propósito de la especificación generada del movimiento humano ha sido el tema de numerosos sistemas notacionales. Las notaciones de baile fueron consideradas como viables, compactas, con modos de expresión registrable en la computadora, para movimiento humano debido a su refinamiento como descripciones de movimiento simbólico. Una animación tenía que tener la herramienta de depuración. La creación directa del movimiento a través de una interfaz numérica o notaciones fue también considerada.

5. Entretenimiento. La gente (o al menos criaturas animadas) es el tema favorito de caricaturas y películas. Las técnicas de animación de 2D fueron las más ampliamente usadas [Cat78]. En un esfuerzo de evitar los sensores en el cuerpo (rotoscoping) de actores vivos, las primeras técnicas de modelado y animación en 3D fueron desarrolladas en la Universidad de Utah, Universidad de estado de Ohio y el Instituto de Tecnología de New York

6. Entendiendo el movimiento. Hay conexiones profundas entre lenguaje natural y movimiento humano. Uno de estos intentos para producir una clase de movimientos observados (sintéticos) caracterizando cambios en una localidad espacial o descripción de orientación sobre el tiempo. Mas recientemente, están produciendo movimiento desde instrucciones o descripciones de un lenguaje natural

Los esfuerzos primarios fueron dirigidos a descripciones de lenguaje del movimiento de objetos. Específicamente, crean representaciones de movimientos de objetos 3D en un camino tal que secuencias de movimiento de imágenes de objetos, podrían ser analizados para producir descripciones de movimientos con oraciones en inglés. Extendiendo el modelo de figuras articuladas, concentrado sobre modelos de figuras humanas validadas gráficamente para ayudar al proceso de entendimiento de imágenes [BOT79]

Las representaciones del movimiento que se desarrollaron, retuvieron una característica críticamente importante: menos metas dirigidas para una especificación de movimiento eficiente. El proceso de metas dadas ha sido desarrollado para satisfacerlas

Un paradigma de simulación fue adoptado y alguno de los problemas especiales de simulación de movimiento humano fue investigado [BSOW78, BOK80]. Mientras otros estudios de locomoción concentrados sobre cinemática inversa para alcanzar metas [Zel82, Gir87, Bru88, BC89]. La solución al problema de administrar múltiples alcances y metas del movimiento que afectan mutuamente muchas partes del cuerpo, lleva a volver a examinar, una restricción basada en posicionamiento y algoritmos generales más robustos para lograr simultáneamente metas múltiples [BMW87]

El estudio en notaciones del movimiento, también llevan a cabo una apreciación de ciertas limitaciones fundamentales: ellas fueron buenas para describir los cambios o resultados finales de un movimiento, pero aun no específicas cuando indican, como un movimiento debe ser realizado. Para justificar las notaciones fue necesario que el realizador (por ejemplo, un bailarín) sea un experto, que tenga el entrenamiento y experiencia para poder hacer la notación del movimiento. La transformación desde notación dentro de movimientos suaves, naturales y expresivos fue parte del arte. La excepción a los sistemas notacionales de fallas cercanamente universales para capturar matices de conducta fueron esfuerzos de notación de forma [BwDL80]. Comenzaron un estudio análogo computacionalmente a la semántica puramente descriptiva de ese sistema. Por un modelo de movimiento humano emergido en 1986 el cual integró la aproximación de cinemática inversa y directa con una aproximación dinámica, modelo basado en fuerzas. Las principales contribuciones a dinámicas basadas en animación se encuentran en las referencias [AG85, Wil87, Hah88].

Las conexiones entre lenguaje y movimiento comenzaron en la mitad de 1970. Se comenzó a investigar el lenguaje natural para el control de animación de tareas. Fueron investigadas nuevas representaciones para el movimiento y técnicas de semántica.

El sistema HIRES, para tareas y procesos de simulación en animación humana. La salida fue producida seleccionando posturas clave predefinidas. Por ejemplo, del famoso problema Filósofos cenando ("Dining Philosophers") usando cinco modelos de figuras humanas, fue producido por simulación de la solución de red petri en el simulador HIRES. En 1989 HIRES fue reemplazado por un nuevo sistema de simulación, YAPS, el cual incorporó planeación temporal con especificaciones imprecisas, interrupción de tareas y tareas de estimación de tiempo basadas sobre desempeño de modelos humanos.

Varios sistemas han usado la notación de conductas para describir y generar movimiento [Zel91]. Se ha usado la notación de conducta en modelos para generar animaciones de manadas de pájaros y cardumen. El trabajo con pájaros y peces usando individualmente un conjunto de reglas simples que les dice a ellos como reaccionar al movimiento de los animales vecinos y a las características del movimiento. Algunos parámetros globales también guían el movimiento de la manada entera. También han descrito un ambiente computacional para simulación de actores virtuales, diseñado principalmente para simular un insecto (una cucaracha en particular) para fines de animación. Muchas de las acciones de la cucaracha esta en caminado y un controlador de

pasos genera el movimiento de caminado. El sistema usa cinemática inversa para posicionar las piernas y se enfoca a posturas de conducta importante.

7. Agentes en ambientes virtuales. Un agente es una representación de una figura humana virtual que se crea y controla por programas realizados en computadoras.

Aunque un aspecto importante de los agentes es su capacidad de comunicación y que puedan colaborar para resolver problemas

Un avatar es un humano virtual controlado por un participante vivo. Puede ser representado visualmente como iconos 2D, personajes, video compuesto, formas en 3D, o cuerpos completos en 3D.

El problema principal se aproxima siguiendo las dimensiones como: apariencia y movimiento, mecanismos de control para interactividad y autonomía, incluyendo detalles, cuidados, locomoción, interacción de agentes múltiples, cooperación y coordinación.

3.2 Movimiento

El concepto de movimiento, es un cambio, en una parte de una figura, sobre un intervalo específico de tiempo. Típicamente, una animación compleja consiste de muchos movimientos distintos y muchos traslapan en cada punto en un tiempo.

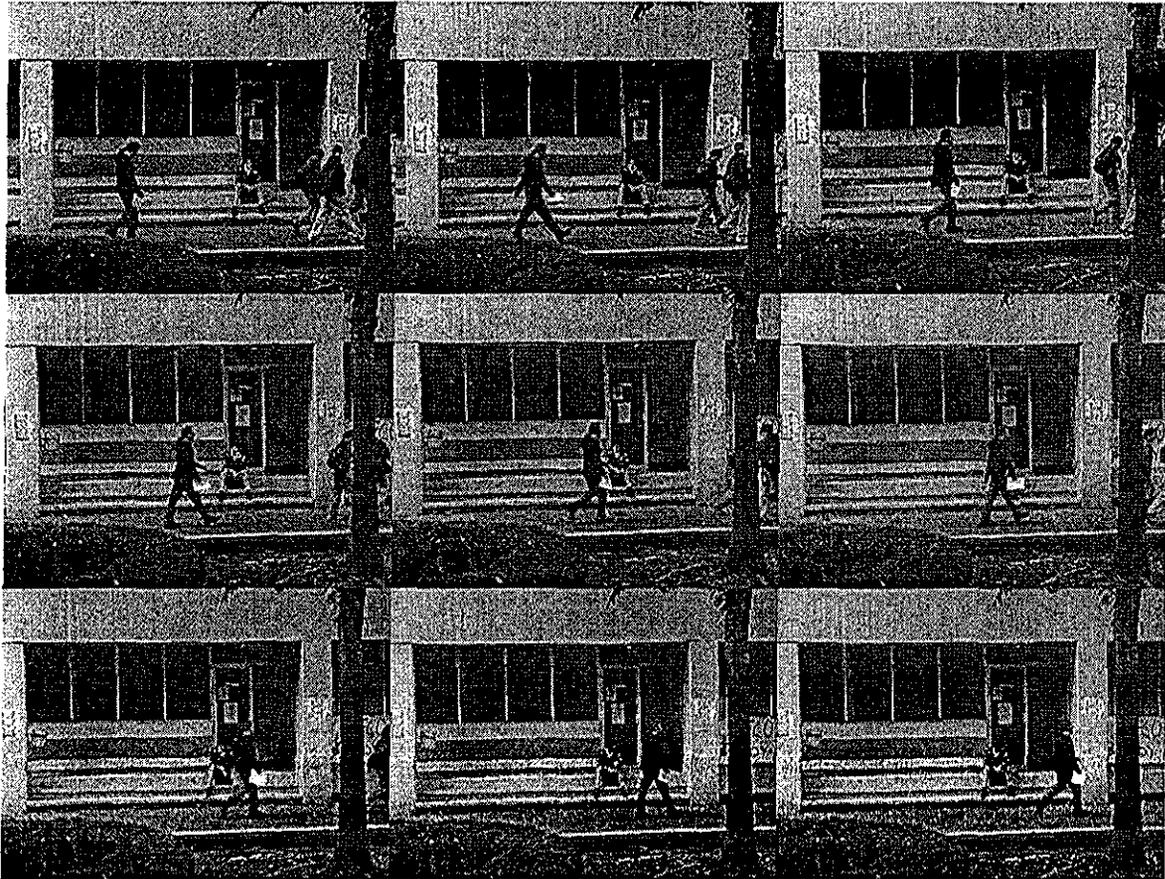


Figura 3.1 Fases del Movimiento

Los movimientos consisten de tres fases distintas, ver figura 3.1. El primer estado de un movimiento es el paso de preacción. Este paso ocurre al iniciar el tiempo del movimiento y prepara la figura para el movimiento inminente, como en el primer cuadro superior izquierdo. El próximo estado es la función misma del movimiento actual, que ocurre en cada intervalo de tiempo después del tiempo inicial hacia el tiempo final, inclusive, como en el segundo cuadro superior. En el tiempo final después del último paso de movimiento incremental, la postacción es activada, como en el tercer cuadro superior. La naturaleza actual de los movimientos y la posibilidad de varios movimientos afectan la conducta de una parte del movimiento, estos tres estados deben ocurrir en cada intervalo de

tiempo en el siguiente orden, preacción, movimiento, postacción. Esto permite que todos los movimientos finales terminen antes de iniciar cualquiera de los movimientos nuevos afectando la misma parte del movimiento.

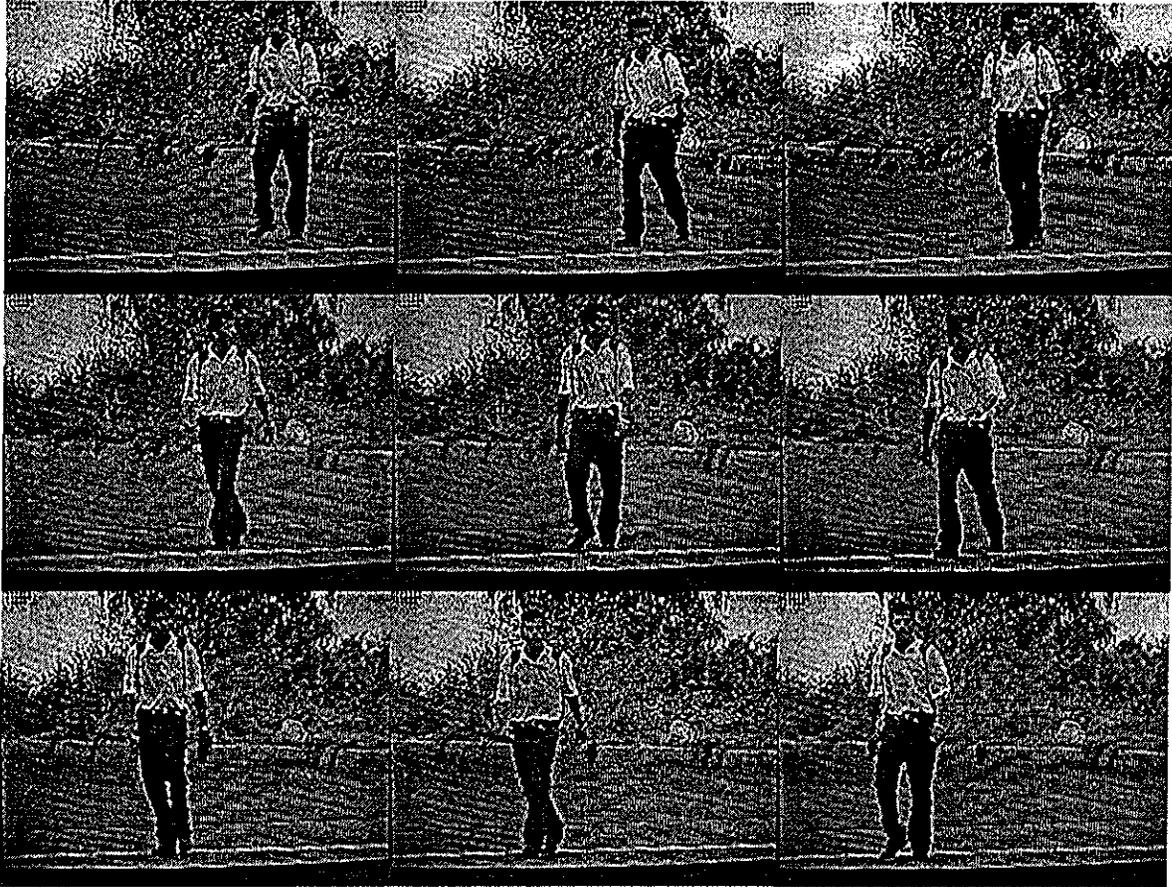


Figura 3.2. Movimientos del Cuerpo. En este modelo real se aprecian los movimientos de sus piernas y brazos al mismo tiempo.

Cada movimiento parte del cuerpo a través de una restricción. Esta es solamente activa cuando el tiempo actual se encuentra entre el tiempo de movimiento de inicio y el tiempo final. Entonces es enteramente posible tener dos movimientos que afecten la misma parte del cuerpo activo al mismo tiempo.

Considere el cuadro superior como configuración inicial, ubique su brazo y pierna tanto el izquierdo y derecho. Observe sus posiciones en un mismo instante de tiempo, ocurren movimientos que afectan por ejemplo del brazo: el brazo, codo y antebrazo al mismo tiempo. Otro ejemplo, de la pierna: su muslo, rodilla, pie al mismo tiempo, ver figura 3.2.



TESIS CON
FALLA DE ORIGEN

Figura 3.3 Secuencias de movimientos del pie.

Los movimientos son combinados para generar secuencias de animación complejas. Un movimiento es mover una parte del cuerpo de un lugar a otro, ver figura 3.3. La posición inicial del movimiento, sin embargo, se define implícitamente en términos en donde la parte del cuerpo está cuando el movimiento inicia. Por ejemplo, una secuencia de movimientos del pie es definida, con un movimiento para cada caída del pie. Cada movimiento sirve para mover el pie de su posición inicial, sin embargo debe estar, cuando el movimiento inicia, a la posición final para ese movimiento.

3.2.1 Caminar Humano

Describir el movimiento del caminado es muy complejo, no sólo hace que los pies tengan que moverse a través del piso, si no que las caderas, espalda, brazos, hombros y cabeza también se mueven en sincronía para mantener al cuerpo en balance. Aunque esos movimientos son complejos, si los separas unión por unión, los mecanismos llegan a ser más claros.

Movimiento coordinado

Las dificultades de añadir al modelado las partes del miembro superior y controlar sus diferentes partes son difíciles para su unión interna en la cooperación de un movimiento y al asignar una unión particular en cualquier animación. Esta dificultad es fácil de demostrar. Ponga su brazo a un lado de forma que la mano tenga la cara hacia arriba, entonces gire la mano para que ésta tenga la cara hacia abajo y continúe girando para que la mano tenga la cara hacia arriba. Intente que este movimiento lo haga involucrando sólo la mano, la muñeca y el antebrazo y luego el brazo y el hombro superior. Agregando al movimiento del torso incluyendo la espina y clavícula, las cuales involucran más grados de libertad hace esta tarea más simple, pero esta se construye en la especificación de ángulos a uniones más complejas. Esto es difícil determinar qué rotación deberá ser asignada en qué tiempo con el fin de modelar su movimiento realista.

Interacciones entre las partes del cuerpo es una preocupación mayor a la de determinar que uniones usar en un movimiento particular. Mientras observando el brazo y la mano en forma separada, simplifica la estrategia de control del sistema independiente, esto relaciona al resto del cuerpo que debe ser tomada en cuenta para un tratamiento de alcance más robusto. Volver a posicionar, girar y doblar el torso, reactivar los movimientos para el otro brazo y un balance contrario por parte de las piernas son partes frecuentes de movimientos que sólo parecen pertenecer a un solo brazo. Esto es casi imposible para una persona alcanzar un objeto y conservar el resto del cuerpo en posición fija. En lugar de extender las uniones a los ejes de sus límites y inducir tensión, otras partes del cuerpo deben cooperar para relevar la tensión del músculo o mantener el equilibrio.

De la misma manera, la manipulación del brazo se usa en muchos movimientos diferentes del cuerpo. Incluso el caminado, que se modela frecuentemente como una actividad de las piernas solamente, involucra el torso, los brazos, incluso la cabeza. Los brazos frecuentemente parecen como una posición útil y simple para empezar la animación de modelado de figuras humanas, pero esto es difícil conservar las tareas en un nivel simple.

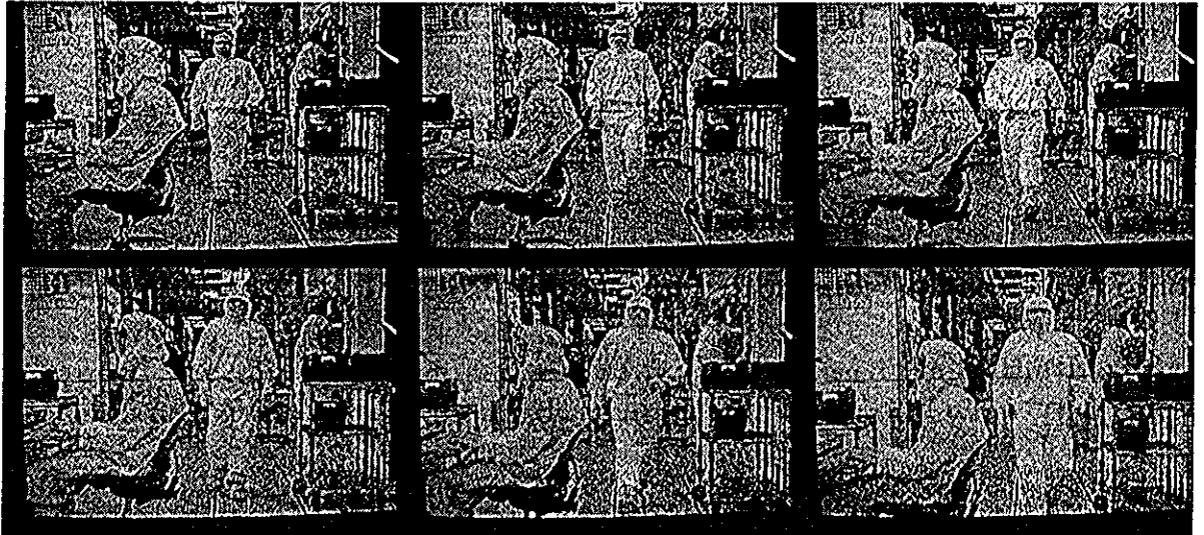
Fases del paso de caminado



Figura 3.4 Fases del Paso del Caminado.

- Fase de giro o de apoyo único. Cuando un pie está sobre el piso mientras el otro gira. Comienza con el momento de despegue, termina con la colisión del pie de giro con el piso y tiene la mayor duración del paso, ver figura 3.4.
- Fase de apoyo doble. Cuando ambos pies están sobre el piso mientras el cuerpo se mueve hacia adelante, ver figura 3.4
- Transiciones entre fases. Al comienzo de la fase de giro está el momento de despegue, el pie está justamente impulsando el cuerpo hacia adelante y la pierna pierde contacto con el

piso La otra transición entre la fase de giro y la fase de apoyo doble se caracteriza por una colisión del pie de giro con el piso, ver figura 3.4.



TESIS CON
FALLA DE ORIGEN

Figura 3.5 Proceso del Movimiento del Caminado.

3.2.2 Proceso de Caminar.

Cada vez que das un paso, avanzas hacia delante inclinando ligeramente el cuerpo y como la pierna diestra atrapa el cuerpo para prevenir a éste de caerse. Si el doblado pone el pie hacia delante, la inclinación doblará sobre la cara. Si mantiene el impulso del pie atrás sobre un paso, caerá plano sobre la cara. Después de que el pie toca el piso, el peso del cuerpo es transferido al pie y la rodilla doblada absorbe el choque. La pierna de frente entonces levanta el cuerpo y impulsa el cuerpo hacia adelante y como la pierna de atrás gira hacia arriba para atrapar el cuerpo otra vez y el ciclo se repite, ver figura 3.5

Al caminar cada parte del cuerpo se mueve desde los pies a la cabeza y cada uno tiene un conjunto único de movimientos.

Caminando.

Llegar caminando, es una de las actividades más comunes que la persona humana emplea. Esta es una actividad compleja que para los humanos es aprendida solo después de un extenso proceso de ensayo y error. Un aspecto que diferencia el caminado al de movimientos de alcance típico, es que este usa las piernas, los brazos y es básicamente cíclico. Mientras su naturaleza cíclica provee alguna uniformidad, componentes acíclicos tales como viraje ocurre periódicamente. En suma, el caminado es responsable para transportar la figura de un lugar a otro y es simultáneamente responsable para mantener el balance. Entonces juegos dinámicos en un rol mucho más integral en la formación del movimiento del caminado.

Un aspecto del caminado que complica su análisis y generación es que este es dinámico. Esto significa que si una figura al caminar emplea una conducta de repente congelada, la figura no está necesariamente en un estado de balance y podría caer al piso. Para fines de animación, esto significa que el movimiento de caminado no puede ser congelado en tiempo y analizado estáticamente para determinar las fuerzas y torques correctas que produce el movimiento. Como un resultado, el conocimiento del movimiento del caminado, en la forma de reunir datos empíricamente o un conjunto de parámetros ajustables por el animador, es típicamente usado como el mecanismo de control global para la conducta de caminado. Atributos como longitudes de grandes pasos, rotación de la cadera, y la colocación del pie pueden ser usados para especificar lo que debería parecer un caminado particular. Un diagrama de transición de estados, o su equivalente, es usado típicamente para transición del paso fase a fase. Cálculos de fuerzas y torques pueden entonces ser añadidos, si desea, para hacer el matiz del movimiento más preciso. Las cinemáticas pueden ser usadas completamente para controlar las piernas, mientras las fuerzas implicadas por el movimiento de las piernas son usadas para afectar el movimiento del cuerpo superior. Alternativamente, las cinemáticas pueden ser usadas para establecer restricciones sobre el movimiento de la pierna tal como la duración del giro de la pierna y la localidad del pie. Entonces las fuerzas y torques necesarias para satisfacer las restricciones pueden ser usadas para resolver los grados de libertad restantes de las piernas. En algunos casos, la dinámica directa puede ser usada después para determinar las fuerzas y torques necesarias para manejar las piernas desde un estado a otro.

3.3 La Raíz de la Figura.

La principal desventaja de modelar una figura articulada como una jerarquía es que un punto sobre la figura debe ser designado como la raíz de la figura. El efecto de la raíz de la figura sobre el algoritmo de cinemática inversa, es que puede solamente manipular cadenas que emanan desde la raíz. La raíz de la figura es un concepto no natural. Esta no tiene analogía natural para una figura en movimiento, con la de una existente, la humana, así esta no tiene lugar en el lenguaje para controlar figuras humanas. Hay varias posibilidades para la raíz de la figura de una figura humana. Para ciertas posturas, hay distintos puntos de referencia sobre la figura que sirven como raíces de buena figura: el pie, el torso más bajo,

y el centro de masa. Muchos sistemas no tienen la habilidad para cambiar la elección de la raíz para localizar esta en el torso más bajo [BKK+85]. Sin embargo, para una postura sentada, el torso más bajo es una buena elección para la raíz. Cuando una figura está parada, los pies son la elección natural para la raíz.

La manipulación de la pelvis provee un control poderoso sobre la postura general de una figura.

El centro de gravedad del cuerpo está en la cadera, todo balance inicia allí.

3.4 Caminado básico paso a paso.

La figura humana en su estado natural tiene restricciones sobre las diferentes partes del cuerpo como sus pies, talones, rodillas, pelvis, centros de masa, manos, codos, cabeza y ojos. Para tener claridad sobre esto se describe que pasa durante el caminado de las imágenes que se obtuvieron de varias personas caminando en el movimiento de cada unión, ver todas las figuras de este capítulo.

3.4.1 La Cabeza y la Espalda.

Si mira la espalda de una persona de lado, puede ver su postura general, que debe ser muy rígida o postura perezosa (figura 3.19) [GM96b], dependiendo sobre la actitud de la persona. La espalda también absorbe algunos de los golpes transmitidos a las caderas desde las piernas, haciendo este doblado desde el frente hacia atrás un poco.

En un caminado estándar, la cabeza intenta permanecer en el nivel, con los ojos apuntando en dirección del caminado, pero su peso alrededor permanece ligeramente balanceado.

3.4.2 Las Caderas, Pelvis, Torso, Espalda, Hombros.

El centro de gravedad del cuerpo está en la cadera, todo el balance inicia allí. Durante un caminado, es mejor pensar en los movimientos de las caderas como dos rotaciones de traslape separadas.

Primero la cadera rota ± 5 grados a lo largo de los ejes de la espalda, hacia delante y hacia atrás con las piernas. Si la pierna derecha está hacia adelante, la cadera derecha rota hacia adelante.

Segundo, en la **posición de postura**, la pierna libre saca la cadera fuera del centro, forzando a las caderas girar de lado a lado, ver la figura 3.20. Estos dos movimientos son entonces transmitidos directos de la espalda a los hombros, que reflejan la cadera para mantener el balance.



Figura 3.6. Rotación de caderas y hombros.

Cuando el pie está totalmente extendido las caderas deben, rotar a lo largo de los ejes de la espalda. Para guardar el balance, los hombros giran en dirección opuesta. (Figura 3.6) [GM96b]. Desde el frente, la espalda está relativamente recta, pero desde arriba, usted puede ver cómo las caderas y hombros rotan en direcciones opuestas para mantener el balance.

En la posición de postura, la vista de frente muestra a la cadera partiendo del centro por el peso de la pierna libre causando una rotación contraria en los hombros. Desde arriba, las caderas y los hombros tienen ángulos igualmente cercanos, ver (figura 3.6) [GM96b]. En la extensión de la segunda pierna, las caderas y hombros son otra vez planos cuando es vista desde el frente. Desde arriba, puede ver la rotación completa de las caderas y los hombros, ver (figura 3.6) [GM96b].

La Pelvis

La pelvis y el torso están estrechamente relacionados. El torso incluye todas las uniones en la espina y rotando estas uniones permite a la figura de sobre doblado. Sin embargo, cuando existe sobre doblado humano, generalmente doblan su pelvis también como su torso. Este manipula insignificadamente las dos uniones de la cadera como una unidad, que puede ser un problema para un modelo en computadora porque no hay puntos fijos únicos bajo las caderas desde el cual rotan. Sin embargo este problema es fácil de manejar controlando la orientación de la pelvis a través de una restricción. La manipulación de la pelvis también activa la conducta del torso. La manipulación de la pelvis provee un control poderoso sobre la postura general de una figura, especialmente cuando combinas el balance y las restricciones del torso. La restricción de la pelvis describe la orientación de las caderas. Los músculos en las piernas hacen que las piernas actúen como una tensión con rotaciones. La cadera y los ángulos de unión proveen solamente un poco más que 90 grados de rotación en la pierna alrededor del eje vertical. Esto significa que la orientación del pie y la orientación de la pelvis están ligadas. Si la orientación de los pies es fija, la orientación de la pelvis es severamente limitada. Si las piernas tienen tensiones con rotaciones, entonces la orientación promedio de la pelvis puede ser determinada promediando simplemente la orientación del pie. Esto parece ser en realidad lo que pasa cuando una persona se para naturalmente: la pelvis esta orientada para auxiliar la fuerza sobre las piernas. La pelvis sigue la conducta de orientación del pie para simular esto.

Al cambiar la orientación global de las caderas, puede mover las caderas hacia adelante o hacia atrás, inclinarlas lateralmente, o girar todo el cuerpo alrededor de los ejes verticales. La manipulación de la pelvis también activa la conducta del torso. Porque de su localidad central, las manipulaciones de la pelvis proveen un control poderoso sobre la postura general de una figura, especialmente cuando combinas con el balance y restricciones del torso vertical guardado. Al rotar la pelvis puede también disparar la conducta del paso activo si la orientación alcanza un ángulo relativamente extremo al pie

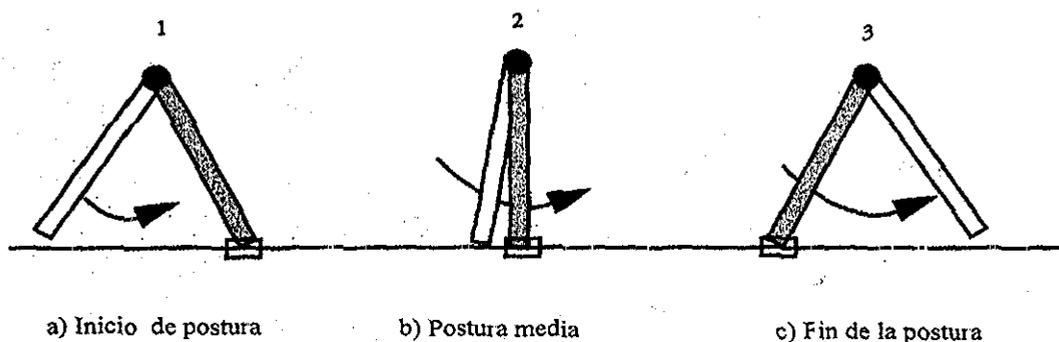


Figura 3.7 Posición de la pelvis durante la fase de postura (plano sagital); la caja indica apoyo de contacto con el piso

TESIS CON
FALLA DE ORIGEN

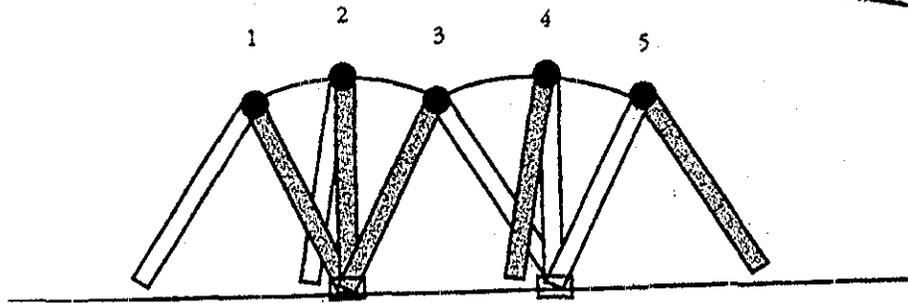


Figura 3.8 Transporte de la pelvis intersectando arcos circulares (plano sagital).

Transporte pélvico.

La pelvis representa la masa del cuerpo superior siendo transportada por las piernas. Usando una representación simplificada para las piernas (figura 3.7) [RP2002], muestra como la pelvis se apoya por la pierna de postura en varios puntos durante la fase de postura del ciclo de caminado. La (figura 3.8) [RP2002] muestra estas posiciones superpuestas durante una zancada total y ilustra el terminado de arcos describiendo el camino básico de la pelvis cuando esta es transportada por las piernas



Figura 3.9 Orientación pélvica durante la fase de postura (plano transversal).

Rotación pélvica.

La pelvis representa la conexión entre las piernas y la estructura en 3D que separa las piernas. La (figura 3.9) [RP2002] muestra la posición de la pelvis en varios puntos durante el ciclo de caminado, como se ve desde arriba. La pelvis gira (rota) cerca de un eje vertical centrado en la postura de la pierna, ayudando a alargar la zancada cuando la pierna de giro avanza a su nueva localidad del pie. Esta rotación de la pelvis sobre la postura de la pierna significa que el centro de la pelvis sigue un arco circular relativo al tope de esa pierna. El tope de la postura de la pierna está rotando sobre el punto de contacto con el piso (figura

3 9) [RP2002], así el camino del centro de la pelvis reensambla una curva sinusoidal, (figura 3 10) [RP2002]

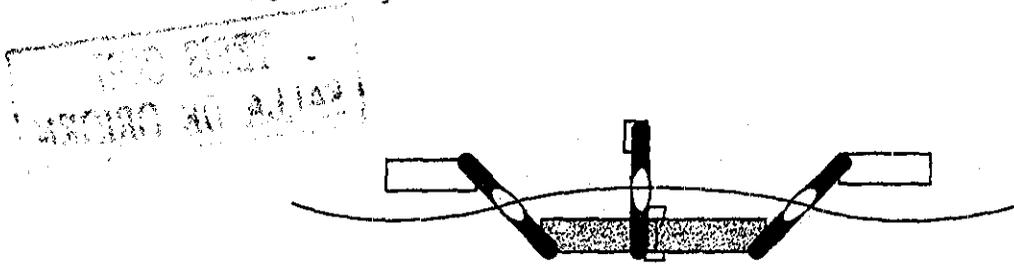


Figura 3 10 Trayectoria del centro pélvico desde arriba (plano transversal), exagerado para fines ilustrativos

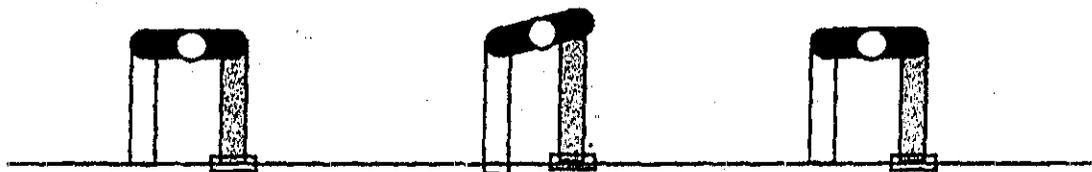


Figura 3 11 Lista pélvica para reducir la cantidad de levantamiento (plano coronal).

Lista pélvica.

El transporte de la pelvis requiere las piernas para levantar el peso del cuerpo cuando la pelvis rota sobre el punto de contacto con el piso, (figura 3.11) [RP2002] La lista pélvica para reducir la cantidad de levantamiento, rotando en el plano coronal.

El Torso

El movimiento del torso de una persona, permite doblar el torso dentro de una postura deseada. Cuando el tiempo de terminación del movimiento sea alcanzado, hay que regresar a la configuración inicial del torso.

La orientación global del torso se consigue con mantener el torso en una orientación vertical. Investigaciones biomecánicas nos dicen que uno de los elementos más constantes en tareas locomotoras humanas simples es la orientación global de la cabeza. Explicando una teoría esta sugiere que la cabeza es el principio sensor de estabilidad. Como la pelvis de la figura rota adelante, atrás, o de lado a lado, el torso automáticamente compensará guardar la cabeza arriba. Entonces la restricción que está sobre el torso más superior, no la cabeza, el cuello es libre de moverse para que la figura mire en ciertos puntos de referencia.

La conducta para mantener la orientación global, involucra todos los tres grados de libertad del torso. Esto permite a otras partes del cuerpo, ser ajustadas mientras la cabeza y el pecho permanecen relativamente fijos. Esto es particularmente importante en ajustes

naturalmente para la pelvis y piernas después de posicionar el torso aceptablemente. Esta conducta no involucra posición, porque esta es usualmente aceptable que tenga la posición variable con respecto al cuerpo.

Los movimientos de la espina son descritos en términos de ángulos de doblado total en las direcciones, lateral y axial. Las técnicas usan factores de peso que distribuyen el ángulo de doblado total a la vértebra individual entre las uniones. Distribuciones de peso diferentes generan doblados diferentes. Estos parámetros son opciones a los controles de conducta del torso porque como ellos gobiernan la conducta del torso, ellos doblan para mantener la orientación apropiada.

Doblando el torso tiende a causar grandes movimientos del centro de masa, así este proceso tiene un gran efecto sobre la postura de la figura en general, particularmente las piernas, por ejemplo, si la figura dobla hacia adelante, las caderas automáticamente correrán hacia atrás así la figura permanece balanceada.

3.4.3 Los Brazos.

A menos que la persona este sosteniendo alguna cosa o haciendo gestos, sus brazos cuelgan flojos en los lados. Cuando caminan, ellos actúan como péndulos, arrastrándolos un poco hacia atrás del movimiento de las caderas y hombros. Aún en la extensión total los brazos deben ser ligeramente doblados en los codos, para hacerlos mirar naturales. El movimiento de los brazos debe involucrar solo las uniones del brazo, ver figuras de este capítulo.

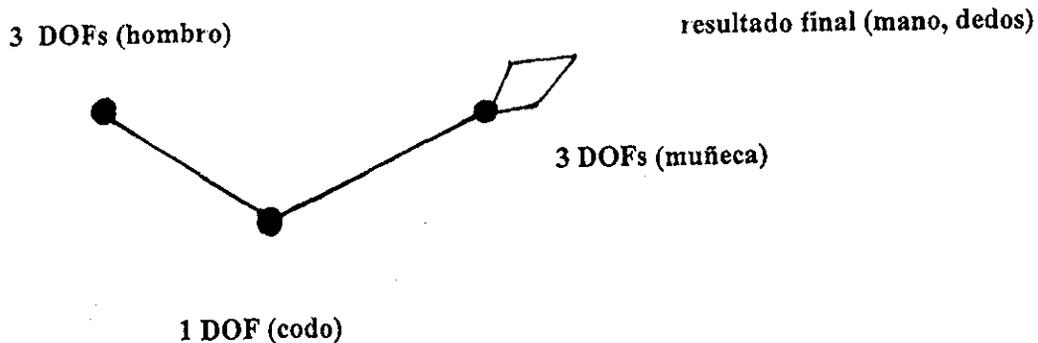


Figura 3.12 Modelo básico del brazo humano.

Modelando el Brazo.

El modelo básico del brazo humano, ignorando las uniones de la mano, puede ser representado simplemente como un manipulador de siete grados de libertad, (figura 3 12) [RP2002]: tres grados de libertad están en la unión del hombro, uno al codo y tres a la muñeca. Una configuración o pose para el brazo es un conjunto de siete ángulos de unión, uno para cada uno de los siete grados de libertad del modelo. La rotación del antebrazo presenta un problema. En la (figura 3 12) [RP2002], la rotación del antebrazo es asociada con la muñeca. Sin embargo, en realidad la rotación del antebrazo no está asociada con una unión localizada como la mayoría de los otros grados de libertad de la figura humana, sino es distribuida a lo largo de los huesos del antebrazo (el radio y cubito) girando alrededor de nosotros. A veces esta rotación está en cambio asociada con el codo, otra aplicación crea una mitad del camino de la unión “virtual” a lo largo del antebrazo para controlar la rotación del antebrazo.

Claro, las uniones del brazo tienen límites, por ejemplo, el codo puede encorvar a aproximadamente 20 grados y puede extenderse a tanto como 160 grados. La mayoría de las uniones se posicionan en alguna parte y raramente logran los límites de rotación de la unión a menos que sean forzadas. Más sutilmente, los límites de unión pueden variar con la posición de otras uniones y se imponen los límites extensos en las uniones para evitar intersección de accesorios con otras partes del cuerpo. Por ejemplo, los tendones hacen más difícil extender la rodilla totalmente cuando uno está doblando a la cadera (el movimiento tocaba uno de los pies). Si se dan fuerza a los límites de unión, algunos movimientos generales pueden obtenerse con éxito usando las cinemáticas directas. Aun cuando un objeto sea llevado por la mano, la cinemática directa junto con el objeto ligado al resultado final crea un movimiento bastante convincente. Pero si el brazo/la mano debe operar relativo a un objeto fijo, como un bulto, la cinemática inversa es necesaria. Hay métodos para prevenir las soluciones hacia los ángulos de unión deseados. Esto ayuda evitar violar los límites de las uniones y produce movimientos más parecidos a los humanos pero todavía falta cualquier base anatómica para producir las configuraciones parecidas como humanos.

Es a menudo útil especificar la posición de la meta, la muñeca en lugar de los dedos para controlar bien las configuraciones producidas. Pero aun cuando la muñeca es fija (resultado final) a una situación deseada y el hombro es similarmente fijo. Hay todavía un número grande de posiciones que podrían adoptarse para satisfacer las restricciones y los límites de unión. Previendo los ángulos y orientaciones de unión con toda seguridad preferible para que ciertas tareas reduzcan el problema de la solución múltiple un poco.

Para controlar el movimiento más precisamente, el usuario puede especificar las posiciones y las orientaciones intermedias para el resultado final tan bien como para las uniones intermedias. Esencialmente, esto establece poses clave para la unión. Pueden usarse las cinemáticas inversas entonces para pasar de una pose a la próxima, puesto que el brazo se conduce todavía a lo largo del camino. Esto proporciona algunas de las economías al usar las cinemáticas mientras le da más control al animador sobre el movimiento final

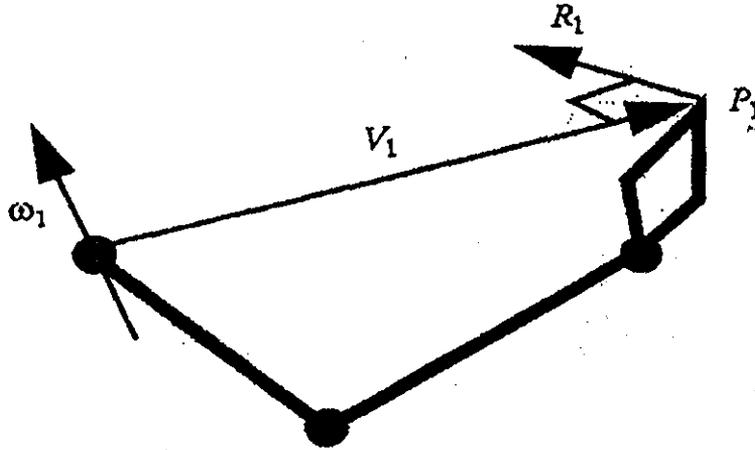


Figura 3.13 Efecto del primer grado de libertad sobre el resultado final $R_1 = \omega_1 \times V_1$.

En el movimiento humano, las uniones más lejanas al resultado final (la mano) la mayoría tienen efecto sobre este. Las uniones más cercanas a la mano cambian los ángulos para realizar cambios finos en la orientación en la última alineación. Esto puede llevarse a cabo por un procedimiento calculando el efecto de cada grado de libertad en el resultado final tomando el producto cruz del eje de rotación, la velocidad angular ω_1 , con el vector de la unión al resultado final, la velocidad lineal V_1 (figura 3.13) [RP2002]. Además, desde que el brazo contiene un grado de libertad un ángulo (codo), un plano entre el hombro, el codo y la muñeca se forma y las posiciones de los brazos preferidas dictan un rango de rotación relativamente limitado para ese plano. Una vez que el plano es fijo, el hombro y ángulos del codo son fáciles de calcular y pueden ajustarse fácilmente sobre ese plano (figura 3.14) [RP2002]. Algún paquete de animación (por ejemplo Maya) le permite al animador especificar una solución de la cinemática inversa basada en un plano y gira el plano como se desee

Algunos estudios neurológicos sugieren que la postura de los brazos sea determinada de la situación deseada del resultado final. Un conjunto de ángulos para el hombro y el codo es calculado desde la posición de la mano y hombro deseada. Finalmente, una orientación de la muñeca es calculada separadamente

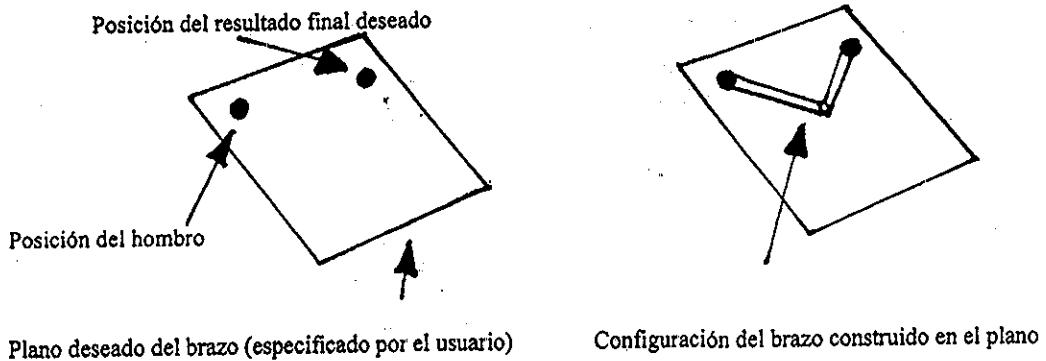


Figura 3.14 Construyendo el brazo en un plano especificado por el usuario.

La unión del hombro.

La unión del hombro requiere una consideración especial. Normalmente se modela como una unión de la pelota con tres grados coincidentes de libertad. El sistema del hombro humano es realmente más complejo. Un modelo más realista de la clavícula, escapula junto con una unión del hombro en tres uniones separadas en un rango limitado proporciona el movimiento muy realista del brazo y hombro. En la (figura 3 15) [RP2002] se proporciona un modelo conceptual de las extremidades superiores

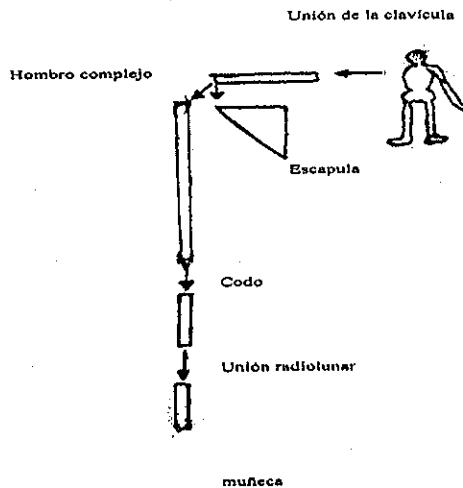


Figura 3.15 Modelo Conceptual de la extremidad superior.

3.4.4 Las Rodillas y los Codos.

Las rodillas y los codos requieren cuidado especial para prevenirlos, desde llegar a ser cerrados desde una extensión total. La posición extendida totalmente no solamente parece torpe, pero esta tiende a causar que el algoritmo de cinemática inversa consiga atraparse en un mínimo local. Porque el algoritmo usa una aproximación de gradiente descendente, si un codo o rodilla alcanza su limite, esta tiene una tendencia a permanecer allí. Para prevenir se usan restricciones que limitan saltos para desalentar las rodillas y codos de alcanzar su valor limite. El salto da un nivel de energía alto al ángulo extendido totalmente. Los saltos pueden ser restringidos en un rango establecido para cualquier ángulo, pero la convención es 10 grados y en la práctica, esta tiende a trabajar bien

Esto mantiene a la figura en su posición de postura natural, una postura más “cómoda”. La literatura en biomecánica describe esto en términos de la fuerza sobre los músculos. La siguiente condición se usa en los codos y rodillas en la postura de una figura articulada: los codos no están “ni doblados ni forzados” y en el nivel medio de apoyo, las rodillas están “rectas pero no tiesas”

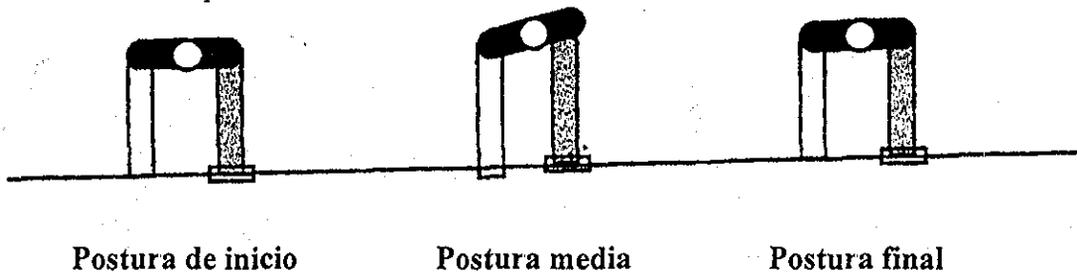


Figura 3.16 Lista pélvica para reducir la cantidad de levantamiento (plano coronal).

Flexión de rodilla.

Como se muestra en la (figura 3.16) [RP2002], en una lista pélvica con piernas en buen estado, las piernas de giro penetrarían el piso. Doblando la rodilla de unión (flexión) permite a la pierna de giro seguramente pasar sobre el piso y evitar el contacto, (figura 3.17) [RP2002]. La flexión en la rodilla de la postura de la pierna también produce algún nivel fuera de los arcos pélvicos producidos por la rotación de la pelvis sobre el punto de contacto con el piso. En adición, la extensión justo antes de contactar con el piso seguido por la flexión de la nueva postura de la pierna en impacto provee un grado de choque de absorción.



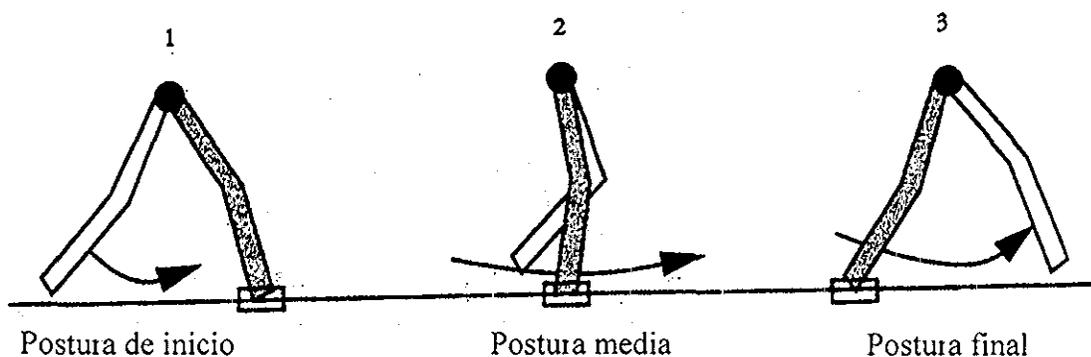


Figura 3.17 Flexión de la rodilla tomando en cuenta la pierna de giro para evitar penetrar el piso, nivelando la trayectoria de la pelvis sobre el punto de contacto y proveyendo alguna absorción del choque (plano sagital).

3.4.5 Los Pies y las Piernas.

Los pies y las piernas impulsan el cuerpo hacia delante. Para mantener el aspecto natural de la figura articulada, se deberá mantener las uniones siempre ligeramente dobladas, aún en una extensión de la pierna total como se puede ver en la Figura 3.18.

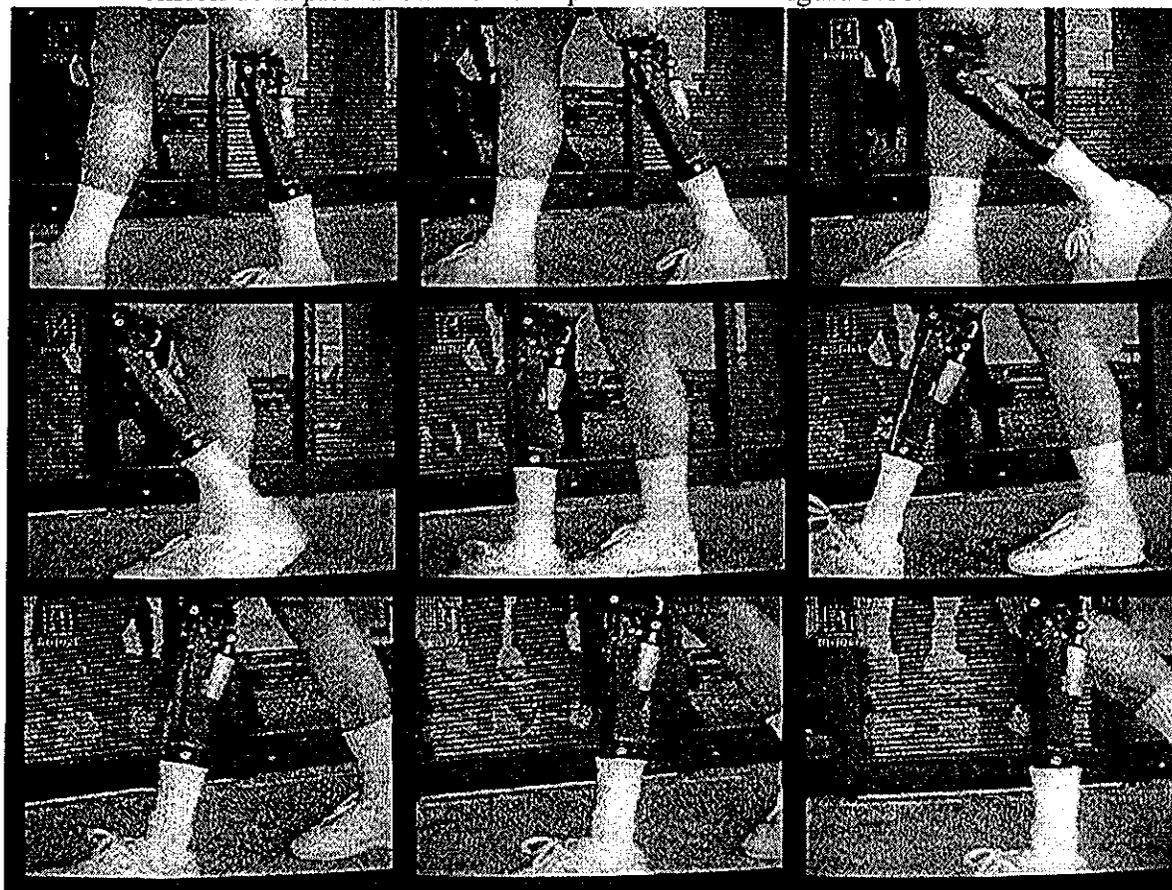


Figura 3.18 Caminado con una rodilla y pie artificial.

Los Pies.

El modelo de una figura articulada tiene un pie con dos segmentos conectados por una unión única del pie y dos restricciones naturales, una sobre los pies y otra sobre el talón. La restricción del pie, pone los pies sobre el piso, la restricción del talón puede poner el talón sobre el piso o permitir a este subirlo si es necesario. Para una figura de postura, se permite guardar el talón sobre el piso y también permitir al talón el control de subir su altura. La conducta del pie es usualmente activada por la manipulación de algunas otras partes de la figura, tal como el centro de masa o la pelvis.

El punto de balance tiene una distribución del peso entre el pie y el corrimiento del peso atrás y adelante. Esta es bien definida a pesar de la posición del pie. Así es el trabajo del punto de balance que sigue el comportamiento del pie.

La orientación del pie durante su manipulación activa, puede involucrar translaciones y rotaciones en el pie, centrado alrededor de los pies. Entonces no es posible rotar y trasladar durante un solo movimiento, la orientación puede ser arbitraria e impredecible o permanente durante el arranque de translación.

La solución que la conducta ofrece es predecir la propia orientación del pie y ajustar la orientación automáticamente cuando el pie es trasladado.

Los pies son controlados a través de restricciones sobre los talones y sobre los pies. Las restricciones del pie controlan la posición y orientación de los pies. Las restricciones del talón controlan solo la altura del talón del piso. La posición del talón, y el pie total, viene de los pies. La selección del pie, el pie izquierdo o el pie derecho, da la posición final del pie. En adición una altura debe ser especificada.

El movimiento causa que el pie se mueva desde su posición inicial a su posición final a través de un arco de cierta elevación. Una altura de cero implica que el pie se mueva en una línea recta. Si ambas posiciones (inicial y final) están sobre el piso, entonces esto significa que el pie resbalara a lo largo del piso. Una altura de 10 cm significa que el pie alcanzará una altura máxima desde el piso de 10 cm a medio camino a través del movimiento. Activa el apoyo de la figura para mover el pie (crea el movimiento). Sin embargo, esto no pasa durante la generación de la secuencia del movimiento.

La conducta del pie depende mucho sobre el apoyo de la figura, aunque el efecto es enteramente sutil y difícil de definir.

Durante una secuencia de movimiento, es mejor tener el apoyo a través del pie sobre el cual la figura tiene más peso.

Esto asegura que este pie permanece firmemente plantado. La conducta del pie puede ser cambiada en un momento específico, para ello es necesario los tiempos de inicio y fin y el tipo de control.

Cuando el tiempo de terminación del movimiento es alcanzado, la conducta regresa al valor previo. El tomar el paso cuando pierdes balance y tomar el paso cuando la pelvis es girada, las conductas reposicionan automáticamente el pie antes que la figura pierda su balance. El pie sin apoyo en este caso es uno de los que llevan menos peso. El pie de apoyo del pie puede ser reposicionado con el fin de mantener el balance.

Si el pie va hacia delante, este automáticamente rota como si la figura estuviera volteando hacia la dirección del pie fijo. La distribución de peso entre el pie permanecerá constante cuando el pie se mueva. Cuando el pie rota, la pelvis automáticamente rota también. El pie de apoyo es siempre muy fijo.

Moviendo los Talones.

Usualmente un movimiento del paso comienza con el talón desprendiéndose del piso. Este movimiento solo por una altura, una altura de cero significa que está sobre el piso.

Una secuencia de pasos involucra varios movimientos de traslape. Este comienza con un movimiento de talón para traer el talón fuera del piso y al mismo tiempo un movimiento de centro de masa para cambiar el peso al otro pie.

Entonces un movimiento del pie causa que el pie se mueva a una nueva localidad. Cuando el pie es cerrado a su nueva localidad, un segundo talón de movimiento causa que el talón sea plantado sobre el piso y un segundo centro de masa cambia algunos de sus pesos de regreso a su pie.

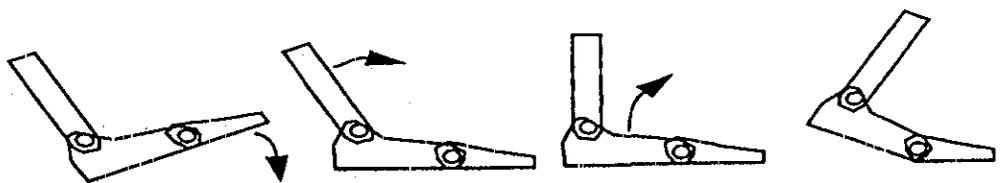


Figura 3.19 Rotación debido a las uniones de tobillo – pie.

Uniones del tobillo y un dedo del pie.

La parte final del rompecabezas del movimiento de caminado es lo complejo del pie, consistiendo del tobillo, de los dedos y el pie en sí mismo. Este complejo comprime varios huesos y muchos grados de libertad y puede ser modelado simplemente (figura 3.19)

[RP2002] , las uniones de tobillo y pie sirven además para enderezar la rotación de la pelvis sobre el pie tan bien como absorber algún choque

3.5 Ciclo de Caminar Humano.



Figura 3.20 Movimiento cíclico de un caminado

Entendiendo la interacción de varias uniones involucradas en locomoción esta el primer paso en entender y modelar la locomoción. El ciclo del caminando es descompuesto en mas detalle, mostrando los movimientos complejos involucrados, ver figura 3 20

TESIS CON
FALLA DE ORIGEN

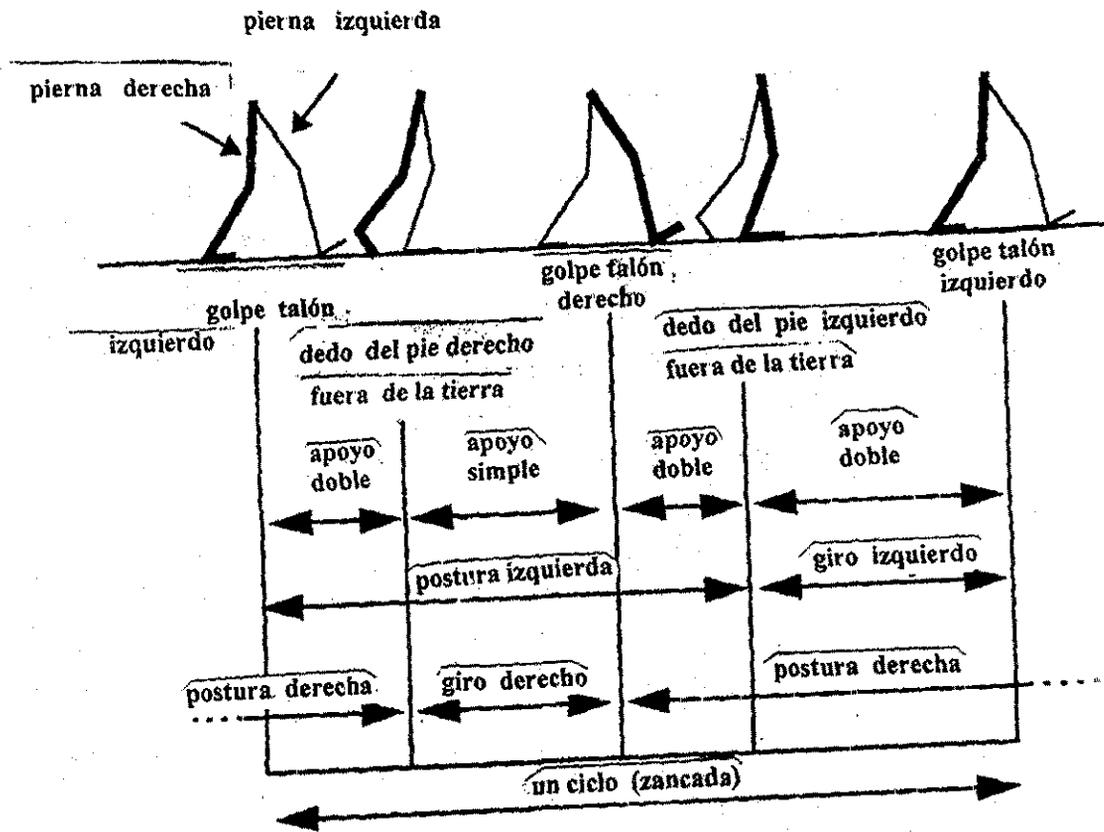


Figura 3.21 Ciclo de caminado.

El ciclo de caminado puede ser descompuesto en varias fases basadas sobre la relación de los pies a sus puntos de contacto con el piso, (figura 3.21) [RP2002]. El caminado a paso largo (zancada) se define por la secuencia de movimientos entre dos repeticiones consecutivas de una configuración del cuerpo. La fase de una postura izquierda de una zancada se inicia con el pie derecho sobre el piso y el talón izquierdo iniciando justamente al golpear el piso. Durante esta fase, el cuerpo se apoya por ambos pies hasta que el pie derecho (su talón se eleva) monta sobre un eje arriba y el dedo derecho del pie deja el piso. La fase de postura izquierda continúa tanto que el pie derecho deja el piso y comienza el giro adelante y como el talón derecho golpea el piso y ambos pies están además una vez sobre el piso. El dedo izquierdo del pie deja el piso y termina la fase de postura izquierda.

La fase de giro derecho es el periodo en la cual el dedo derecho del pie deja el piso, la pierna derecha gira adelante y el talón derecho golpea el piso. Note que la fase del giro derecho es un subintervalo de la fase de postura izquierda. El final de la fase de giro derecho inicia la fase de postura derecha y fases análogas ahora proceden con los roles de la pierna izquierda y la pierna derecha intercambiadas. El ciclo de caminado se caracteriza alternando periodos de apoyo simple y doble.

Tiempo de Caminado Humano.

En pasos de un caminado normal, un paso toma 1/3 o 2/3 de un segundo (8 a 16 poses) y en promedio 1/2 segundo por paso. Un ciclo es un segundo.

Las cinemáticas del caminado.

La animación de la pierna puede ser realizada por un control apropiado de los ángulos de unión. Como se menciono previamente, un ciclo de caminado de la pierna se compone de una fase de postura y una fase de giro. La duración de la fase de giro es el tiempo entre el contacto del piso al levantar el dedo del pie hasta el golpe de talón. La aproximación más básica de generar el movimiento del caminado es para el animador especificar una lista de ángulos de unión para cada grado de libertad involucrado en el camino. Hay varias fuentes para datos empíricos describiendo la cinemática de varios caminados con varias velocidades, (figuras 3.22 a 3.26) [RP2002], son graficados los ángulos sobre el tiempo para la variedad de uniones involucrados en ciclo de caminado, tan bien como dar valores para el desplazamiento lateral de la pelvis.

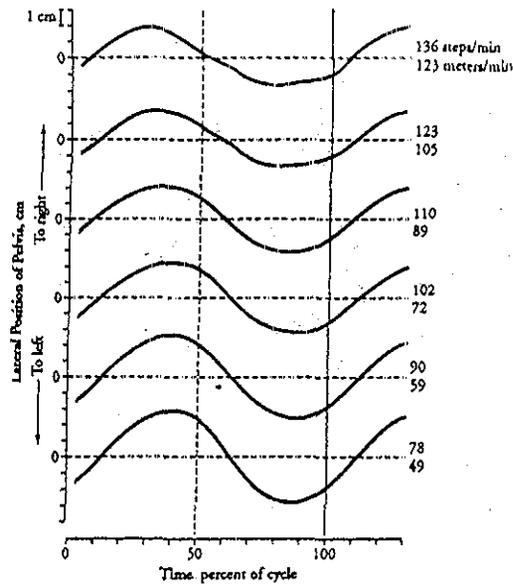


Figura 3.22 Desplazamiento lateral de la pelvis.

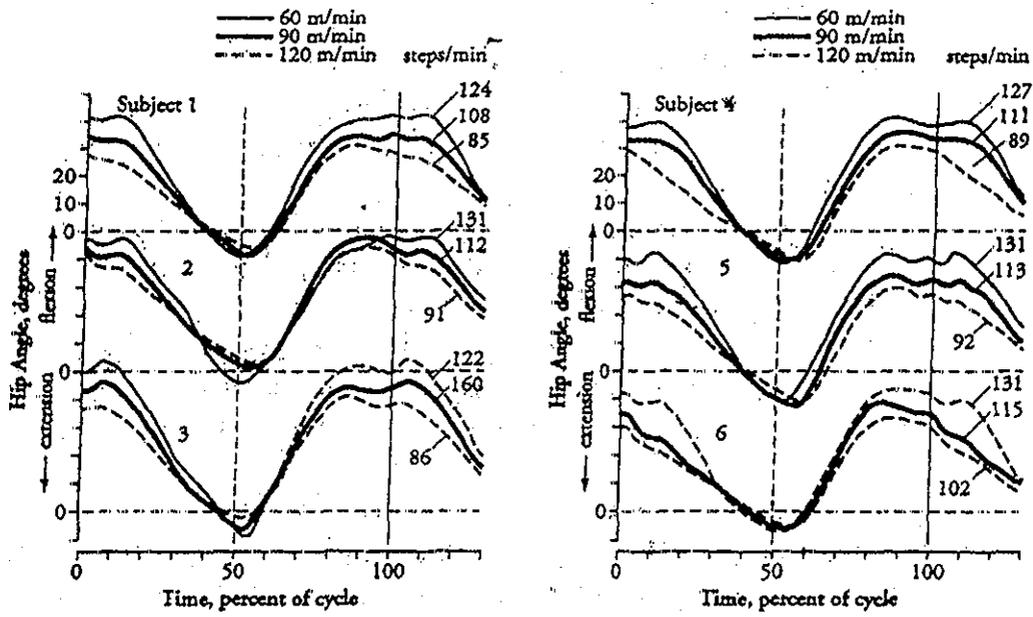


Figura 3.23 ángulo de la cadera.

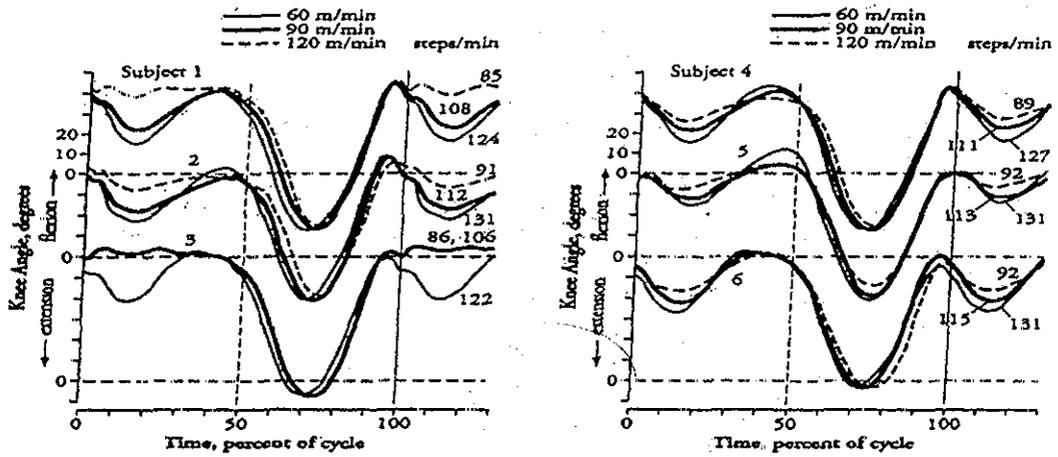


Figura 3.24 ángulos de la rodilla.

25

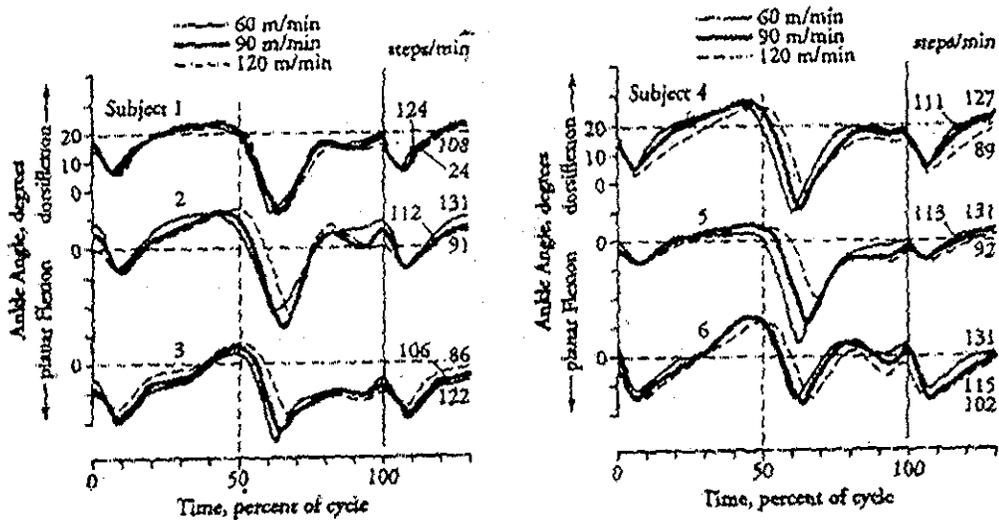


Figura 3.25 ángulos del tobillo.

TESIS CON FALLA DE ORIGEN

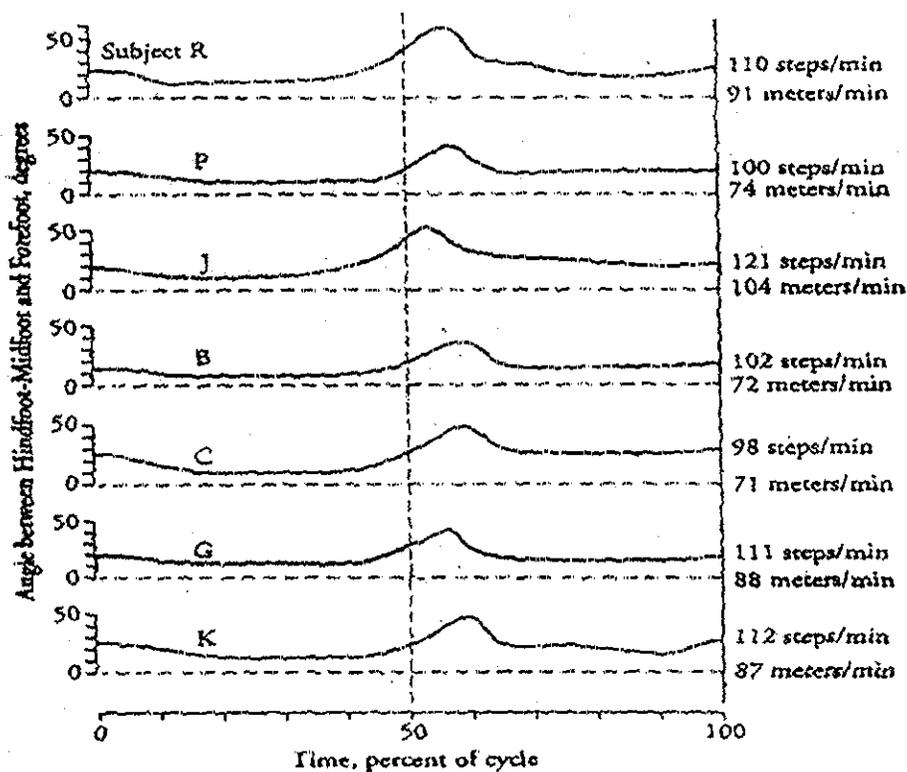


Figura 3.26 ángulos del pie

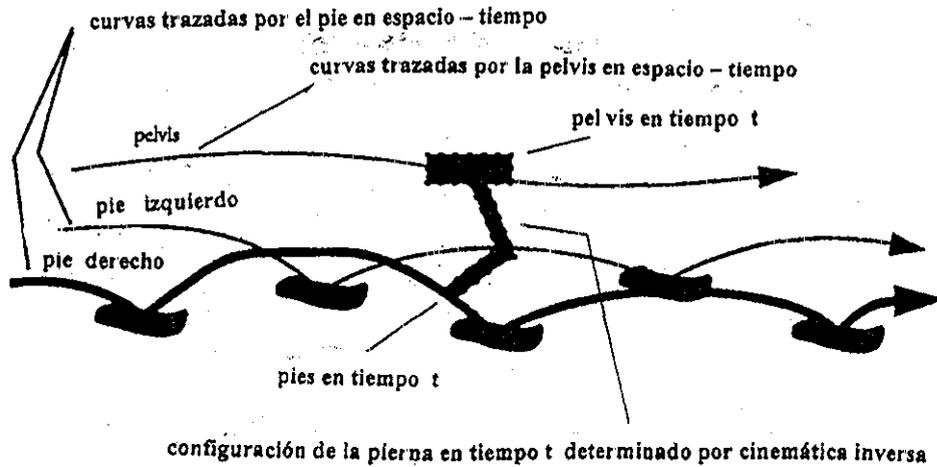


Figura 3.27 restricciones satisfechas para la pelvis y pies de la cinemática inversa.

Es una tarea onerosa para el animador especificar los ángulos en todas las uniones. En adición, esto lleva a una habilidad artística para diseñar valores que creen caminados únicos que desvíen en cualquier camino precisamente al coleccionar sus datos. Cuando se crean nuevos caminos, el animador puede especificar valores cinemáticos tales como movimiento pélvico, localidad del pie y trayectorias del pie. La cinemática inversa puede ser usada para determinar los ángulos de las uniones intermedias. Construyendo las curvas en el espacio en tiempos trazados por la pelvis y cada pie, el usuario puede determinar la posición para cada frame dado en la animación. Entonces cada pierna puede ser posicionada considerando la pelvis fija cuyo apéndice ligado a una posición del resultado deseado es la posición correspondiente sobre las trayectorias curvas del pie, (figura 3.27) [RP2002]. Sensitivamente a las longitudes del segmento pueden producir configuraciones que disminuyen dirigir el pie en contacto sólido con el piso durante el caminado. La cinemática inversa es también útil para forzar a mantener la propia localidad del pie.

TESIS CON
FALLA DE ORIGEN

Capítulo 4.

Métodos de Cinemática y Dinámica para la Especificación del Movimiento.

Este trabajo requiere de la comprensión de técnicas de animación por computadora y de métodos para crear y controlar los movimientos de figuras articuladas. Un esqueleto puede animarse variando las rotaciones locales aplicadas a cada ángulo. También debe aplicarse al ángulo de la raíz una translación global. La especificación del movimiento y el problema del control es eso de dirigir la manera en que estas transformaciones cambian con el tiempo. Hay dos acercamientos fundamentales en general, a este problema: la cinemática y dinámica. Aquí se presentan los métodos de cinemática y los métodos dinámicos en general para la especificación del movimiento, los tipos de control disponible para cada uno y las aplicaciones de éstos en la animación de figuras articuladas. Un sistema de animación de figuras articuladas adecuadas es el resultado de incorporar todas las técnicas discutidas aquí en algún grado. Este tema se basa sobre el material [AG85], [BN88], [Hah88], [IC88], [SJ99], [Wil87], [Wil91], [WCH88].

4.1 Métodos de la Cinemática.

Habiendo modelado la figura articulada con los segmentos y uniones, se necesita ahora tratar el problema de como manipularlo. Hay dos problemas básicos: dados todos los ángulos de la unión, como calcular la configuración espacial y recíprocamente, dada una cierta postura, qué valores deben asignarse a los ángulos de la unión. El primer problema, la cinemática directa, simplemente es una cuestión de multiplicación de matrices de transformación, el segundo problema, la cinemática inversa, es más difícil de resolver. En un ambiente de manipulación interactiva, el usuario solo se preocupa por la configuración espacial, que al transformarse en los ángulos de la unión; se realiza la cinemática inversa

para una aplicación deseable. La cinemática inversa determina el movimiento del mecanismo, que es una técnica común en ingeniería mecánica, particularmente en robótica. En los robots, sin embargo, se evitan normalmente los grados de libertad redundantes. Es más, el cálculo se lleva a cabo normalmente en los sistemas particularmente de unión. En contraste, un objeto interesante en el dominio de la animación en computadora, el cuerpo humano, tiene muchos grados de libertad redundantes. Por consiguiente se estudian los métodos de cinemática sobre todo en el contexto de límites de unión y grados de libertad redundantes.

4.1.1 Antecedentes.

La interpolación para keyframes es bien establecida en los sistemas comerciales para animación. Se han propuesto varios métodos de interpolación que proporcionan varios grados de control sobre una trayectoria y variaciones en la velocidad a lo largo de ella. La técnica de interpolación basada en spline cúbica de Hermite, proporciona tres de los parámetros: continuidad, tensión y prevención. Que se proveen para controlar la longitud y dirección de los vectores tangentes a la trayectoria en los keyframes. Si se modifica la dirección de los vectores tangentes, controla la curva cuando atraviesa un Keyframe. Cambiando la longitud de los vectores tangentes cambia el valor interpolado del keyframe y así controla la velocidad. La animación tradicional se logra con las escenas apropiadas de estos parámetros. Desgraciadamente, los tres parámetros de la spline afectan la curva. En el método de interpolación doble una trayectoria la define una spline cúbica, que atraviesa una serie de valores keyframe y una spline adicional controla la trayectoria curva. Esto proporciona el control sobre la velocidad. Y para que un animador logre el perfil de velocidad deseado a lo largo de la trayectoria este requiere de un proceso de ensayo y error. Sin embargo, no es posible encontrar una solución analítica para las curvas arbitrarias y uno debe acudir a la aproximación de métodos numéricos.

En el caso de Cinemática Inversa, el sistema Critter permite la manipulación en un esqueleto para crear keyframes. Badler propuso un algoritmo de cinemática inversa, para aplicarlo en las partes múltiples del cuerpo, durante la manipulación del esqueleto [BMW87] y un rango para los límites de los ángulos en la solución de cinemática inversa. El sistema PODA y el controlador de pasos de Sims usan la cinemática inversa para generar el movimiento de la pierna en modelos para esqueletos. En estos sistemas, una fase de planeación determina las colocaciones y trayectorias del pie. El algoritmo de cinemática inversa genera el ángulo y orientación de las piernas, así como de los pies que se mueven a lo largo de las trayectorias, entre cada apoyo del pie.

4.1.2 Cinemática Directa.

Usando cinemática directa, la posición de cualquier objeto dentro de un esqueleto sólo puede controlarse indirectamente especificando los ángulos en las rotaciones entre la raíz y el propio objeto. La cinemática directa se involucra poniendo la posición y orientación de objetos explícitamente en los momentos de frames específicos. Para los esqueletos, esto significa poner los ángulos directamente en las rotaciones seleccionadas y aplicar la translación global al ángulo de la raíz, para crear una pose. Para evitar hacer esto en cada frame de una animación, una serie de poses keyframe pueden especificarse en los diferentes frames, las poses intermedias se calculan interpolando los parámetros de ángulo entre los keyframes. La figura puede animarse entonces desplegando cada pose intermedia. La interpolación lineal entre keyframes es el método más simple para generar estas poses intermedias, el movimiento resultante es poco satisfactorio normalmente. Las discontinuidades de las primeras derivadas, en los ángulos interpolados entre los keyframes, dan una desigualdad. El uso de métodos de interpolación de orden superior, como el de splines, puede proporcionar velocidad y aceleración continua y por lo tanto transiciones más suaves entre y a través de keyframes.

La interpolación produce a menudo valores intermedios que no reúnen realmente los requisitos del animador, algunos controles sobre el proceso de la interpolación son cruciales. Los valores interpolados para un solo grado de libertad sobre el curso de una animación forman una trayectoria curva. La forma de la trayectoria y del movimiento del objeto, son dependientes de los valores del keyframe y del tipo de interpolación spline usada. Un editor interactivo que le permita al animador ver y modificar la forma de una trayectoria puede ser una herramienta útil. Una vez que una trayectoria se define, la calidad del movimiento puede ser modificada más allá variando la proporción en qué la trayectoria es cruzada.

Para la animación del esqueleto, la interpolación del keyframe no trabaja bien; los pocos ejemplos buenos de animación de figura keyframed son más un tributo a la habilidad y paciencia del animador que la conveniencia de la técnica para la tarea. Una dificultad mayor, pueden ser los problemas etiquetados "grados de libertad": Para los esqueletos interesantes, hay absolutamente demasiados grados de libertad que deben mantener sus valores; El nivel de detalle requerido del animador para especificar una sola pose es excesivo. Intentando controlar el movimiento interpolado, posiblemente modificando manualmente, ciento de trayectorias curvas puede ser tedioso. Mientras es esencial que el animador tenga algún control al nivel del ángulo, los niveles más altos de control son deseables, para especificar los movimientos coordinados de grupos de ángulos. La estructura jerárquica del esqueleto también causa los problemas. El único ángulo que un animador puede posicionar explícitamente es el ángulo de la raíz en la jerarquía; las posiciones de todos los otros objetos en el esqueleto dependen de las rotaciones a los ángulos anteriores. Esto lo hace difícil de dar fuerza a las posiciones restringidas cuando creas una pose keyframe. Por ejemplo, si la raíz de la jerarquía para un bípedo está en la pelvis, entonces poniendo un pie en el suelo y guardándolo hay algunos problemas; si el pie

ya está en el lugar, entonces una curvatura en la rodilla moverá el pie, que debe entonces reposicionarse modificando la rotación en el ángulo de la cadera. La habilidad de reestructurar la jerarquía sobre un nuevo ángulo de la raíz sólo es marginalmente útil. En nuestro ejemplo, haciendo el apoyo del pie, la nueva raíz de la jerarquía, permitiría una curvatura de la rodilla que deja el pie en el lugar. Sin embargo, esto también moverá el resto del cuerpo, que puede mover otro segmento del cuerpo previamente posicionado, como el otro pie. Esto lo hace dando fuerza, a las restricciones de posiciones múltiples, un proceso frustrante. Aun si un animador se ha asegurado que se posicionan ambos pies correctamente en una serie de poses keyframe, no hay ninguna garantía que interpolando las rotaciones del ángulo mantengan las posiciones correctas del pie en los frames intermedios. Es bastante común ver las sucesiones interpoladas del keyframe para figuras donde los pies parecen penetrar a través de, o resbalar adelante del suelo. Esto puede remediarse especificando los keyframes adicionales, espaciando keyframe más pequeños en el proceso, entonces la animación empieza a parecerse al posicionamiento frame by frame de animación tradicional. Esto derrota el propósito entero de interpolación, que se piensa releva al animador del aburrimiento, de especificar el movimiento, en una base frame by frame. Mientras la cinemática directa combinada con un esquema de interpolación simple puede bastar para animar los objetos simples, realmente no depende de la tarea de animar las figuras articuladas.

4.1.3 Cinemática Inversa.

Las técnicas de cinemática inversa proporcionan el control directo sobre la colocación de un resultado final del objeto, resolviendo los ángulos para las rotaciones que ponen al objeto en la posición deseada. Debe estar claro que la cinemática inversa ofrece una alternativa atractiva al girar los ángulos individuales explícitamente dentro de un esqueleto. Un animador puede especificar directamente la posición de un resultado final, mientras el sistema calcula automáticamente los ángulos que necesita de cada parte. Sin sorpresa, el problema de la cinemática inversa se ha estudiado extensivamente en el campo de la robótica, aunque sólo es recientemente que las técnicas han sido adoptadas para la animación en computadora. La cinemática inversa proporciona el control sobre las jerarquías de ángulos que mueven los miembros de un esqueleto y se pone mucho más manejable. Sin embargo, a menudo el método subyacente para el movimiento generador confía todavía en métodos estrictamente de la cinemática. Desgraciadamente los métodos de la cinemática no producen el movimiento convincente sin una cantidad considerable de esfuerzo en la parte del animador. Los métodos de cinemática directa y inversa, no producen el movimiento con la clase de integridad dinámica que nosotros esperamos de nuestra experiencia con las leyes físicas del mundo real.

4.2 Métodos Dinámicos.

La animación basada en simulación dinámica es atractiva, porque el movimiento generado adhiere leyes físicas, proporcionando un nivel de realismo que es sumamente difícil de

reproducir con los métodos de la cinemática. Para el análisis dinámico, las descripciones del objeto deben incluir atributos físicos tales como el centro de masa, la masa total, los momentos y productos de inercia. Aunque hay muchas ecuaciones de movimiento, todas son esencialmente equivalentes a $F = ma$, que relaciona la aceleración de un objeto de masa m que tiene en respuesta a una fuerza F aplicada al centro del objeto de masa m . El movimiento generado por la simulación física se controla por la aplicación de fuerzas y torques que pueden variar con el tiempo. Pueden categorizarse técnicas para el control del movimiento dinámico como métodos dinámicos directos o los métodos dinámicos inversos. La distinción esencial entre los dos está en las fuerzas básicas y torques (suma de todos los momentos M_i de las fuerzas respecto de un punto dado) que manejan el movimiento.

4.2.1 Antecedentes.

Dinámica Directa Una buena introducción y apreciación global de los elementos de simulación dinámica directa para animar los cuerpos rígidos puede encontrarse en [Wil91]. Simular el movimiento de poliedros de objetos rígidos respondiendo a colisiones, se presenta por Hahn [Hah88]. En el problema de inestabilidades numéricas las ecuaciones de movimiento para los esqueletos articulados son inherentemente mal condicionados, independientemente de su formulación. Se presenta el mal condicionado cuando el esqueleto asume una postura en pequeños cambios incrementales, producto de un grado de libertad y aceleraciones grandes en otra parte; así las técnicas de solución numéricas tienen dificultad. Maciejewski [Mac90] contiene que estas situaciones frecuentemente ocurren para las figuras articuladas y es inherente en la estructura de la mayoría de los esqueletos. Las ecuaciones mal condicionadas no sólo tienen implicaciones para el análisis dinámico, sino también para los algoritmos de cinemática inversa. Uno de los esfuerzos pioneros para controlar una figura puramente articulada a través de dinámica directa, fue la simulación del sistema Virya de Wilhelms [Wil86]. En particular, informes de Wilhelms dicen que el acoplamiento de las ecuaciones dinámicas se hace difícil y no controlan la figura. Otros esfuerzos simulan problemas similares de esquemas de movimiento y usan dinámica directa [AG85] [WCH88].

Dinámica Inversa Barzel and Barr [BB87] hicieron uso de dinámica inversa para el modelado. Forsey and Wilhelms [FW88] han usado la dinámica inversa para manipular un esqueleto articulado en un sistema de interpolación de cinemática tradicional. El sistema Manikin durante la interacción con la figura realizó el análisis dinámico, usando la fórmula recursiva de Armstrong [AG85] para las ecuaciones de movimiento. Una meta posicional podría especificarse interactivamente para una parte del cuerpo. El sistema Manikin calcula las fuerzas para empujar la parte del cuerpo hacia la meta. Esto permitió la manipulación de la figura de una manera similar a la manipulación de la cinemática inversa. El sistema Dínamo de Isaacs [IC88] combina las restricciones de cinemática con la dinámica inversa en el keyframe. Sin embargo, Isaacs en un esfuerzo más ambicioso al esquema de animación realiza la simulación de una marioneta tradicional, controlada por las varas y cordones unidos a los miembros. Mientras técnicamente impresionante, este ejemplo señala la necesidad por los buenos métodos de control sobre el esquema del movimiento.

dinámicamente simulado Brotman and Netravali [BN88] proponen una aproximación dinámica inversa obligando a dar fuerza a restricciones de cinemática de keyframe Sin embargo, la solución incorpora una restricción adicional en la energía ejercida por éstas fuerzas finales. Girard [Gir91] ha aplicado técnicas de optimización restringidas para determinar la distribución de velocidad. Las notas criterio de optimización de Girard tiene un efecto significativo en la calidad percibida de movimiento Resolviendo para un perfil de velocidad que minimiza el gasto de energía, un movimiento oscilante relajado para el miembro, como para alcanzar un objeto produce el movimiento de tal meta y dirige las tareas Witkin and Kass [WK88] proponen un método síntesis del movimiento, ellos lo llaman "restricciones de espacios de tiempo" que demuestran ser capaces de sintetizar las trayectorias y el cronómetro de movimientos para las figuras articuladas simples. La investigación está dirigiendo las limitaciones del método interactivamente Badler ha usado dinámica inversa basado en restricción para sintetizar las trayectorias de miembros cargadas con la tarea de mover una carga entre dos posiciones diferentes Las trayectorias se calculan incrementalmente y está restringido por las medidas de fuerza, confort y ejercicio La naturaleza reiterativa del algoritmo difiere fundamentalmente de la solución global encontrada por los métodos de optimización En cambio, un conjunto de heurísticas biomecánicas que son intentadas para imitar el proceso, por el cual las personas mueven las cargas, se usa para guiar el proceso de la solución El método produce con éxito trayectorias del miembro que logran la tarea

4.2.2 Dinámica Directa.

La dinámica directa involucra aplicación explícita de fuerzas y torques variantes en tiempo a los objetos. Un poco de fuerzas, como estas debido a la gravedad y colisiones entre los objetos, pueden manejarse automáticamente para el sistema de animación; otras fuerzas son directamente aplicadas por el animador a los objetos en la escena Se aproxima el movimiento controlando una serie de pasos discretos a tiempo y a cada paso, resuelve las ecuaciones de movimiento, para aceleración de un objeto, en respuesta a las fuerzas aplicadas. Dada la posición y velocidad de un objeto del paso anterior a tiempo, la aceleración se integra para determinar una nueva velocidad y posición, respectivamente, dos veces para el paso de tiempo actual Extendiendo este acercamiento está desafiando a la simulación de esqueletos articulados Habrá una ecuación de movimiento general, para cada grado de libertad en el esqueleto Esto lleva a un sistema grande de ecuaciones que deben resolverse por los métodos numéricos, el gasto computacional es considerable La formulación adoptada para representar las ecuaciones de movimiento afecta significativamente el costo del método de solución Una solución para la formulación basada en la matriz de Gibbs-Appell, para el ejemplo tiene la complejidad $O(n^4)$ para n grados de libertad [Wil87]. Armstrong ha propuesto una formulación recursiva, alternativa que reduce la complejidad a $O(n)$ [AG85], permitiendo realizar las simulaciones dinámicas de estructuras articuladas simples sobre tiempo real. Pero la simulación dinámica razonablemente compleja no puede realizarse en esqueletos articulados en general a las velocidades interactivas en las máquinas de un solo procesador, aunque la formulación recursiva puede ser bastante rápida para ser tolerable. Complicando la materia el hecho es

que las ecuaciones de movimiento para los esqueletos articulados son considerablemente más complejas que para objetos simples, estas deben incluir las condiciones para modelar las interacciones entre las partes del cuerpo conectadas. Este acoplamiento de ecuaciones dinámicas controla hechas extremadamente difíciles, desde el movimiento de un segmento del esqueleto, las fuerzas y torques ejercerán en los segmentos adyacentes; **la noción que el movimiento del esqueleto puede controlarse adecuadamente aplicando torques al ángulo individualmente es incorrecta.** Los esfuerzos para neutralizar normalmente esta propagación no deseada de torques involucran poniendo saltos a tiempo y apagadores (intercambio) a cada ángulo para mantener una orientación deseada. Desgraciadamente, este tipo de control invariablemente lleva a un conjunto difícil de ecuaciones que causa la inestabilidad severa en técnicas de solución numéricas. Para que la fuerza y keyframes del torque, especifiquen los momentos diferentes sobre el curso de sucesión del movimiento entero; se usaron splines cúbicas. Durante la simulación dinámica, los perfiles de fuerza/torque se probaron y combinaron con las fuerzas debido a las colisiones y gravedad, determinando la fuerza instantánea y las medidas del torque para el paso de tiempo actual. El uso de interpolar curvas, es conceptualmente similar a la cinemática de interpolación de keyframe, aproximación directa descrita previamente. La diferencia es que el movimiento no se maneja directamente por las curvas interpoladas, pero indirectamente a través de las ecuaciones de movimiento. De hecho, la simulación dinámica directa está mejor preparada para tareas que pueden proponerse como los problemas del valor inicial. Es decir, tareas para que perfile las posiciones iniciales, velocidades y la fuerza/torque, son conocidas a priori y la meta es generar el movimiento resultante. Esta formulación puede ser satisfactoria para animar escenas de objetos inanimados simples dando volteretas realísticamente y haciendo botar a través de un ambiente, pero no solicita la animación de tareas específicas. Por ejemplo, simulando una pelota que hace botar en un suelo es simple hacerlo, dada una altura inicial y velocidad; la necesidad de la simulación sólo considera la fuerza de gravedad y reacciones a las colisiones con el suelo, para generar el movimiento convincente. Sin embargo, si la meta es tener el salto de la pelota tres veces y aterrizar en una taza el problema es mucho más difícil. La posición inicial exacta y velocidad de descarga de la pelota lo aterrizará en la taza, son difíciles de determinar. Todavía precisamente es la clase de problema que aparece en la animación; el animador sabe qué el movimiento debe ocurrir, pero no sabe las condiciones iniciales de antemano y los perfiles de fuerza/torque necesarias para producir el resultado deseado.

4.2.3 Dinámica Inversa.

Métodos dinámicos inversos determinan automáticamente las funciones fuerza y torque necesarias para verificar el estado de una meta. En el caso degenerado, la meta declarada es una descripción completa del movimiento y el objetivo es determinar las fuerzas y torques que reproducen el movimiento bajo la simulación dinámica directa. **Mientras este caso es de interés en la robótica, su aplicación es de uso pequeño en un sistema de la animación; después de todo, si las trayectorias del movimiento y el cronometro son de antemano conocidos el gasto de la simulación física es innecesario.** Métodos recientes

más interesantes son los que permiten especificar restricciones relativamente de alto nivel o metas y entonces calcula las fuerzas y torques necesarias para encontrar las metas.

4.2.3.1 Restricciones Geométricas.

Se definió un modelo como una colección de objetos relacionada por las restricciones geométricas. Existen algunas restricciones sencillas útiles para el modelado, incluso las restricciones del punto a punto para ligar dos objetos juntos, restricciones del punto a un camino para mover un objeto a lo largo de un camino predefinido y restricciones de la torcedura para controlar la orientación de un objeto. Las restricciones fueron usadas para introducir fuerzas y torques en una simulación de la dinámica directa del modelo. Estas restricciones de fuerzas y torques actúan para mover al modelo hacia un estado en que todo las restricciones sean satisfechas. La imposición de restricciones de posición en las partes del cuerpo era cumplida aumentando la masa de partes restringidas artificialmente, con el sistema que constantemente calcula las fuerzas adicionales necesarias para mantener la parte en su lugar, cuando las otras partes se mueven. Las sucesiones del movimiento podrían ser generadas guardando el estado del cuerpo de los diferentes puntos durante el análisis dinámico y usando estos estados guardados después como el keyframe para interpolación de la cinemática. La aproximación para la fuerza final tomada aquí, de convertir todas las restricciones en fuerzas y torques que dirigen el movimiento durante el análisis dinámico, tiene sus limitaciones. Las fuerzas finales son a menudo los modelos como los saltos a tiempo simuladas y apagadores (intercambios) que entregan una fuerza proporcional a la velocidad del movimiento. Este difícil método de control es vulnerable en el sistema resultante de ecuaciones y por las oscilaciones indeseables sobre los puntos de satisfacción de restricción. Escogiendo apropiadamente los coeficientes para las restricciones son a menudo una cuestión de ensayo y error. En contraste con la aproximación de fuerza final, varias formulaciones para las ecuaciones dinámicas de movimiento pueden incluir las ecuaciones de restricciones explícitas. En su lugar causando la introducción de fuerzas adicionales en la simulación, las restricciones de la cinemática se usan para quitar los grados de libertad al sistema, en cambio ellos especifican algunas de las aceleraciones implícitamente en el sistema. Las aceleraciones restantes sin restricciones de grados de libertad puede resolverse entonces. El método de solución asegura que las fuerzas reaccionan debido a restricciones de keyframe que son introducidas dentro, debido a la solución para los grados de libertad sin restricciones. Esto permite restricciones de la cinemática para especificar el movimiento en algunas partes del esqueleto, mientras las otras partes sin restricciones reaccionan en forma real al movimiento prescrito. En casos donde todas las partes están restringidas, la técnica se reduce a una aproximación del keyframing simple. Está aproximación ilustra una mezcla interesante de simulación dinámica con el control de la cinemática.

4.2.3.2 Sin Restricciones Geométricas.

Considere el problema dinámico inverso de mover una masa del punto de la posición A a la posición B en un intervalo de tiempo dado t . No hay ninguna función única de fuerza sobre el intervalo t que lo logrará, este sistema debe escoger entre aplicar una fuerza grande por un período corto de tiempo, o aplicando una fuerza más pequeña sobre un período más largo, ambos métodos pueden lograr la meta de alcanzar la posición B keyframe en el momento t . Este problema no sólo determina lo que tiene que ocurrir, en este caso mueve A a B, sino también cómo el movimiento tiene que ocurrir. Se han propuesto varios métodos que intentan describir la calidad de movimiento sin restricciones geométricas en la solución dinámica inversa. Estas aproximaciones están basados en técnicas bien establecidas por perfeccionar las funciones sujetas a un conjunto de restricciones. El problema se formula, como resolver para el conjunto de fuerzas de restricción, que minimicen la energía, encontrándose las restricciones impuestas, por los valores de cinemática de keyframe.

Estos métodos de optimización restringidos asumen que los caminos del movimiento completos o parciales para los miembros son de antemano conocidos, e intenta derivarlos poniendo las mejores fuerzas y torques al movimiento del miembro a lo largo del camino. El problema fundamental, el sintetizar las trayectorias del miembro, de estos pasos en primer lugar para el movimiento coordinado. Este uso de optimización restringida parece prometer particularmente y es capaz de obtener de un productor complejo, el movimiento coordinado, físicamente correcto con pocas entradas de un usuario. Sin embargo, la aproximación produce sistemas muy grandes de ecuaciones que deben resolverse y no puede ser considerado útil para la animación de la figura interactiva.

3.3 Controles del Movimiento

Producir un algoritmo para una amplia gama de sucesiones del caminado realistas, requiere de un híbrido para el control de movimiento cinemático y dinámico.

4.3.1 Antecedentes.

Zeltzer[Zel82] era un defensor pionero de la necesidad para el control de alto nivel sobre las figuras articuladas. Construyendo en el trabajo de Zeltzer, Bruderlin [BC89] desarrolló una estrategia del control jerárquico similar para las sucesiones generadoras de caminado bípedas, pero con simulación dinámica incorporada para derivar el movimiento, además de la pierna confiando en los datos del sensor que registra movimientos (rotoscoped).

3.3.2 Control del Movimiento Cinemático y Dinámico.

Los esfuerzos de la investigación bosquejados arriba son intentados para proporcionar los niveles más altos de control del movimiento sobre cinemática y dinámica. La meta es poder especificar los movimientos al nivel de la tarea y producir el movimiento. Dado el estado actual de estos esfuerzos, parece que pasará algún tiempo antes de que los sistemas sean capaces de sintetizar el movimiento para lograr las tareas arbitrarias. El sistema es responsable para descomponer las descripciones de la tarea de alto nivel ("camina a la puerta" o "alcanza la taza"), en las primitivas de movimiento de bajo nivel y para la coordinación de estas primitivas. Las primitivas de bajo nivel pueden consistir en keyframes para la interpolación, metas de la cinemática inversa, simulaciones dinámicas directas, metas dinámicas inversas restringidas, o una mezcla de todas estas aproximaciones. Zeltzer describe una estrategia del control para sintetizar las sucesiones de caminado en un esqueleto. Se descomponen las instrucciones de caminado de alto nivel, en un conjunto de programas para control del motor (MCP) que maneja los movimientos de miembros individuales o ángulos. La estrategia del control es basada en una máquina del estado finito responsable para activar y dejar fuera de funcionamiento el MCPs apropiado en los momentos apropiados. Los MCPs de Zeltzer's consisten en valores de ángulo de cinemática obtenidos de sensores que registran movimiento (rotoscoped) de caminados humanos y así era puramente la cinemática. No obstante el sistema demostró la utilidad del concepto.

El usuario puede instruir un esqueleto para caminar a una velocidad particular y puede especificar la frecuencia deseada del paso y longitud del paso. Estas instrucciones se descomponen dinámicamente basadas en el nivel más bajo MCPs que maneja el movimiento abstracto, pone el par de piernas y rodillas. Los MCPs realizan interpolación dinámica de un conjunto de keyframes de la cinemática esencialmente para los movimientos de la pierna durante el ciclo del caminado. Los valores y espaciado de la cinemática de keyframe, se derivan de los parámetros de entrada, combinando el conocimiento sobre modelos de la locomoción humanos (literatura biomecánica). Las fuerzas y torques que manejan el movimiento del mecanismo ambulante simplificado son iterativamente ajustados hasta que los ángulos de solución keyframe se logren en los momentos correctos. Se realiza entonces la cinemática en esqueletos de piernas articuladas hacia el mecanismo subyacente. El algoritmo puede producir una gama amplia de sucesiones de caminado realistas y es un verdadero híbrido del control de movimiento cinemático y dinámico. La decisión para usar un modelo dinámico simplificado puesto a punto para caminar específicamente parece legítima, el sistema resultante de ecuaciones es pequeño, relativamente estable y barato de resolver. Una aproximación similar se ha usado, para construir un algoritmo saltador basado en la simulación de un modelo de masa y saltos a tiempo subyacente simple. Desgraciadamente, el control de alto nivel proporcionado por los algoritmos de esta naturaleza viene al gasto de generalidad; cada estrategia de control debe ponerse a punto para un movimiento específico. Pero desarrollar una estrategia de control tal es difícil. Derivando las ecuaciones para simular la dinámica del mecanismo

subyacente requiere un poco de sofisticación matemática. En la ausencia de un algoritmo de cinemática inversa, el método Bruderlin [BC89] de trazar el movimiento del modelo dinámico subyacente al movimiento del esqueleto puede proponer problemas al implantador. A una magnitud grande, el éxito de las estrategias del control anteriores es debido a la naturaleza predecible, repetitiva de locomoción. Las estrategias del control de alto nivel desarrollado para las sucesiones de movimiento arbitrarias todavía parecen una meta distante.

7
7
1
10
15

Capítulo 5.

Diseño y Control del Caminar de la Figura Articulada.

El entendimiento de la cinemática y la forma de emplear su solución en algunas partes del cuerpo para diseñar y controlar su movimiento, ayudan a proporcionar un algoritmo de cinemática para aplicarla en la animación de una figura en particular. Este capítulo trata formalmente su problema, la aproximación más común y aproximaciones diferentes para resolverlo. El tema de esta capítulo se basa en material [Bad99], [MA92], [SJ99].

5.1 Antecedentes

Las cinemáticas estudian el movimiento independiente de las fuerzas subyacentes que producen su movimiento. Incluyen posición, velocidad y aceleración, esto es todas las propiedades geométricas y tiempo relacionadas al movimiento [MA92]. La representación de cinemática más común en robótica es la notación de Denavit y Hartenberg. Esta representación deriva un conjunto de parámetros para describir una conexión basada sobre mediciones entre los ejes de un manipulador. El objetivo con estas notaciones cinemáticas en robótica es desarrollar una representación estándar que los investigadores puedan usar en los análisis y descripción de manipuladores. Hay varios tipos de manipuladores que son extremadamente comunes en la comunidad de investigación robótica. La adopción de una representación estándar, simplificará grandemente el proceso de analizar e implementar algoritmos robóticos, entonces así los algoritmos son descritos en la literatura usando estos manipuladores. En el proceso de construir un algoritmo de cinemática inversa se incluye la iteración y el jacobiano inverso. El método pseudo inverso fue introducido a la comunidad de computadoras gráficas por Girard 1985 para su generador de pasos PODA. Girard explotó la redundancia de extremidades de un animal en un intento de minimizar ángulos que violan límites. Las capacidades de cinemática inversa de SIMS's para controlar pasos y el sistema Chadwick's Critter son también basados sobre

esta técnica Ninguno de estos esfuerzos sugiere soluciones específicas a los problemas que el método exhibe alrededor de singularidades, puesto que parece razonable asumir que cada uno de estos sistemas no desempeñará bien aproximaciones de configuraciones singulares Badler y Zhao aplican un procedimiento de optimización métrica variable, para proveer un control interactivo sobre una postura de figuras articuladas Los límites de rango en ángulos se presentan como restricciones al optimizador y un número de funciones objetivo para restricciones geométricas simples son desarrolladas Estos incluyen, por ejemplo, restricciones punto a punto, restricciones punto a plano, restricciones en orientación y otras de una naturaleza similar Restricciones simultáneas sobre múltiples partes del cuerpo pueden ser impuestas, sumando simplemente las funciones objetivo individuales minimizadas para cada parte restringida Esto es una herramienta interactiva útil que permite la manipulación de cinemática inversa de una figura, manteniendo un conjunto de restricciones sobre el cuerpo. Phillips describe una función objetivo que intenta un balance en la figura Jack[Bad99], como ejemplo el método puede ser extendido para manejar arbitrariamente metas complejas no geométricas, este trabaja mejor sobre estaciones de trabajo gráficas de alto desempeño Badler admite refrescamiento de la pantalla periódicamente con soluciones intermedias desde el optimizador a retener iterativamente, aun después de los posicionamientos en la pantalla que solamente la solución final es útil y que las soluciones intermedias no deberían estar considerando movimientos Una técnica más simple de minimización es la de gradiente conjugado usada por Alt and Nicolas[Bad99], que provee manipulación de cinemática inversa y animación de extremidades, animando metas de posición sobre el tiempo En contraste a la aproximación de Badler ellos desempeñan minimización sin restricciones, eligiendo respetar límites de ángulos, o sea valores que solucionan los límites del ángulo variable Esto tiene también valor, notando que algún intento antes por Badler, de solucionar el problema de cinemática inversa estuvo basado sobre algoritmos heurísticos simples [BMW87] Estos métodos no aparecen haber ganado una amplia aceptación; Badler ha abandonado desde entonces estas aproximaciones a favor del método de optimización descrito arriba

5.1.1 Condiciones de la Figura Articulada y en Robótica en la Animación.

Es importante dar algunas definiciones de términos que son particulares a la animación de figuras articuladas Además, ya que muchos resultados de robótica han puesto a prueba su utilidad en este campo también se incluyen varios términos de robótica.

Las cinemáticas. Estudio del movimiento, esto es todas las propiedades geométricas y tiempo - relacionadas del movimiento

La figura articulada. Una figura articulada es una estructura que consiste de una serie de eslabones rígidos conectados a las uniones. Aunque otras uniones existen en la robótica, la animación de la computadora normalmente se restringe a las rotaciones o las uniones rotatorias.

Los grados de libertad (DOF). Los grados de libertad (DOFs) son iguales al número de posiciones variables independientes necesarias para especificar el estado de una estructura en una figura articulada.

El resultado final. En la mayoría de las manipulaciones industriales le llaman cadenas abiertas y al final libre de una cadena abierta de eslabones se llama el resultado final.

El vector de estados. El vector de estados de todas las posibles configuraciones de una figura articulada, es un conjunto de parámetros independientes que definen las posiciones, orientaciones y rotaciones de todas las uniones que constituyen la figura. Una configuración particular en una figura, se describe por el vector de estados:

$$\theta = (\theta_1, \dots, \theta_N)$$

La dimensión del vector de estados es en general igual a los grados de libertad de la estructura articulada. Por ejemplo, cualquier cuerpo rígido sin restricciones tiene seis grados de libertad, tres translaciones y tres rotaciones. Así su vector de estados es:

$$\theta = (x, y, z, \mu, \phi, \varphi)$$

En estas condiciones una animación de una figura articulada se reduce a encontrar un camino de $N -$ dimensiones, o grados de libertad en su vector de estados.

5.2 El Problema de Cinemática Inversa.

Un esqueleto puede ser modelado como una colección jerárquica de objetos rígidos conectados por uniones. Un manipulador se refiere a una cadena cinemática de segmentos dentro de un esqueleto y se supone que las uniones conectando segmentos dentro de esta cadena son uniones de revolución rotando sobre un solo eje. Un extremo del manipulador, es la base que esta fija y no puede moverse; el otro extremo distante de la cadena se mueve libremente. El resultado final se incorpora en las coordenadas del marco de la unión más distante en la cadena y la posición del resultado final es un punto dentro de éste marco y la orientación del resultado final se refiere a la orientación del marco mismo. En cada unión en la cadena, una unión variable determina una transformación M entre los dos marcos de coordenadas adyacentes compartiendo la unión. La transformación M_i en una unión de rotación i que es una concatenación de una translación y una rotación, de las cuales ambas son relativas a las coordenadas del marco de uniones padre i . Eso es,

$$M_i = T(x_i, y_i, z_i)R(\theta_i) \quad (5.2.a)$$

Donde $T(x_i, y_i, z_i)$ es la matriz que traslada por desplazamiento de unión i desde su unión padre $i-1$ y $R(\theta_i)$ es la matriz que rota θ_i sobre la unión i del eje de rotación. La relación en la cadena entre cualquiera de los dos sistemas coordenados i y j se encuentra concatenando las transformaciones en las uniones encontradas durante una transversal desde la unión i a la unión j :

$$M_i^j = M_i M_{i+1} \dots M_{j-1} M_j \quad (5.2.b)$$

Puesto que la posición y orientación del resultado final, con respecto al marco base, se encuentra simplemente concatenando las transformaciones en cada unión en el manipulador. Dado un vector (θ) de uniones variables conocidas entonces el **problema de cinemática directa** de calcular la posición y orientación del resultado final del vector x , es una materia simple de concatenación de matrices y se describe en la sección 5.2.1.

5.2.1 Cinemática Directa.

El movimiento de todas las uniones se especifica explícitamente por el animador. El movimiento al resultado final (por ejemplo las manos y los pies de una figura articulada) es determinado indirectamente por la acumulación de todas las transformaciones que llevan a ese resultado final, como la estructura del árbol es 'descendido'. Éste, en el caso de la pierna de una figura articulada, sería el efecto combinado de las transformaciones a la cadera, rodilla y tobillo. La Cinemática Directa dado θ , se deriva X [MA92]. Es decir la función:

$$X = f(\theta) \quad (5.2.1)$$

Donde, X es de dimensión n y θ de dimensión m . Cuando la complejidad de la articulación incrementa, cada suma, de una unión agrega en la jerarquía, un grado de libertad en la figura y una dimensión en el vector de estados. Así el animador tiene más y más transformaciones cada vez que controlar para lograr una animación más expresiva, puede demostrar también ser complicado e intrincado para lograrlo en la práctica. El animador puede usar un idioma llamado scripting, para ayudar a reducir la complejidad y trabajo, una biblioteca de animaciones preespecificada (para las tareas bien - definidas) y para la animación del ciclo del paso de la figura, repetidamente. Pero si la meta es colocar el resultado final X en una posición y orientación especificada, entonces para lograr la meta, determinando apropiadamente los ángulos de las uniones variables del vector (θ) , requiere una solución inversa, descrita en la sección 5.2.2.

5.2.2 Cinemática Inversa.

A veces llamado 'movimiento dirigido a la meta', el animador sólo define la posición de los resultados finales. La cinemática inversa resuelve la posición y la orientación de todas las uniones en la jerarquía del eslabón que lleva al resultado final. La Cinemática inversa dado X , se deriva θ [MA92]. Es decir la función:

$$\theta = f^{-1}(X) \quad (5.2.2)$$

En el caso de cinemática inversa, cuando n incrementa, el problema de encontrar para un X dado, se vuelve **bajo definido** y se dice que el **sistema es redundante**. Un subespacio entero θ definido por:

$$\theta(\theta_1, \dots, \theta_n) \in \theta_x$$

Si X mapea a $f(\theta)$, el problema se aproxima agregando restricciones al sistema al reducir el subespacio de posibles soluciones. Cada restricción forma un subespacio y la intersección de éstos subespacios de restricción con θ_x nos da el juego de posibles soluciones. Se dice que la diferencia entre el número de grados de libertad y el número de restricciones es el grado de redundancia. El criterio de restricción típico puede incluir la minimización de energía, conservación de velocidad adquirida y así sucesivamente. Considere la (figura 5.2) [MA92], que le muestra un manipulador de dos – eslabones, en dos configuraciones, las dos satisfacen la restricción que el resultado final se posiciona en X . Si especificamos la unión entre los dos eslabones a ser un codo en estructura articulada humana, entonces una solución sería imposible y el subespacio de la solución por consiguiente es dividida en dos. En robótica, los manipuladores se dicen tener una enorme 'área de trabajo diestra' en el plano a través de las uniones. Un sistema que controla a un manipulador tiene que escoger una sola solución y esto podría lograrse minimizando la cantidad que cada unión tiene que mover para alcanzar una nueva posición X de una posición actual Y . Generalmente, un conjunto de ecuaciones no lineales parecen salir de cualquier sistema y una solución de la cinemática inversa tiene que hablar de las soluciones existentes de los tres problemas, las soluciones múltiples y el método usado. En robótica se ha mostrado que todos los sistemas con las rotaciones y las uniones prismáticas, teniendo un total de seis grados de libertad en una sola serie de cadenas, son numéricamente solucionables. En la animación de la computadora de estructuras articuladas, el método de la cinemática inversa parece atractivo. Por ejemplo, digamos que nosotros estamos interesados en hacer a un actor sintético, que camine a una cierta velocidad y encima de un terreno particular alcanzar una posición dada. Un animador debe ser capaz de construir este proceso usando comandos de alto nivel para movimiento de meta dirigida, los cuales pueden ser realizados por el método de la cinemática inversa. El movimiento dirigido a la meta está restringido por el funcionamiento del algoritmo de la cinemática inversa en donde simplemente llena los componentes del vector de estados. El animador no tiene ningún control sobre estos componentes. **Los hechos son que la cinemática inversa se vuelve computacionalmente más cara cuando la complejidad de la articulación aumenta y**

que es muy difícil usar cuando tiene la intención de especificar una animación natural o particular, que hasta ahora, sólo ha sido útil en un subconjunto restringido de la animación en una computadora general. Se usa en las aplicaciones donde la jerarquía maneja, parte de la animación del resultado final del movimiento. Tales animaciones incluyen el caminado y el posicionamiento del brazo y la mano. La solución del problema de cinemática inversa no es tan simple. La función f es lineal mientras haya un mapeo único desde θ a X en la ecuación (5.2.1). Lo mismo puede ser dicho para el mapeo inverso de la ecuación (5.2.2). Cuando f no es lineal puede tener muchos ángulos θ para una posición X particular. La aproximación más directa para resolver el problema sería obtener una solución de forma cerrada de la ecuación (5.2.2). Pero las soluciones de forma cerrada pueden solamente ser derivadas para un conjunto restringido de manipuladores con características específicas y aun estos resultados tienen que ser resueltas en un conjunto de ecuaciones no lineales. Una solución analítica general para manipuladores arbitrarios no existe, en su lugar el problema debe ser resuelto con métodos numéricos para resolver sistemas de ecuaciones no lineales. Los métodos de solución más común están basados sobre inversión de matrices o técnicas de optimización.

5.3 Resolver la Velocidad para Controlar el Movimiento.

Desde entonces la naturaleza no lineal de la ecuación (5.2.2) la hace difícil de resolver, una aproximación natural es hacer lineal el problema sobre la configuración actual del manipulador – entonces la relación entre velocidades angulares y las velocidades lineales del resultado final es

$$X' = J(\theta) \theta' \quad (5.3.a)$$

La relación lineal dada la matriz del jacobiano es

$$J = \delta f / \delta \theta \quad (5.3.b)$$

El cual mapea cambios en los ángulos variables θ a cambios en la posición y orientación X en el resultado final. J es una matriz de orden $(m \times n)$, donde n es el número de ángulos variables y m es la dimensión del resultado final del vector X , que tiene o tres dimensiones para una tarea de posicionamiento simple, o seis dimensiones para una tarea más general en la posición y orientación. La columna i esima de J representa el cambio incremental X_i en la posición (y orientación) del resultado final resultante desde un cambio incremental en el ángulo variable θ_i . Invertiendo la relación de (5.3.a) provee las bases para la velocidad resuelta que controla el movimiento

$$\theta' = J^{-1}(\theta) X' \quad (5.3.c)$$

Si la inversa de J es conocida, podemos calcular cambios incrementales en los ángulos variables que producen un cambio incremental deseado en la posición y orientación del resultado final. Un esquema iterativo simple para resolver el problema de cinemática inversa puede estar basado sobre la ecuación 5.3 c

5.3.1 El Jacobiano.

El **Jacobiano** es la extensión de diferenciación multi-dimensional de una sola variable. El Jacobiano J es la matriz de orden $(m \times n)$ de derivadas parciales que relacionan cambios diferenciales en θ ($d\theta$) a los cambios diferenciales en X (dX):

$$dX = J(\theta) d\theta \quad (5.3.1a)$$

Dónde el elemento (i, j) del **Jacobiano** J se obtiene de:

$$J(i, j) = \delta f_i / \delta \theta_j \quad (5.3.1b)$$

Para ello se divide el elemento diferencial de tiempo para dar:

$$X' = J(\theta') \theta' \quad (5.3.1c)$$

Dónde X' es la velocidad del resultado final. Que para la mayoría generalmente es un vector de seis dimensiones. Que incluyen la velocidad lineal V y la velocidad angular Ω y θ' es el tiempo derivado del vector de estados

El **Jacobiano**, entonces mapea las velocidades en el vector de estados a las velocidades en el espacio cartesiano. Estas dos cantidades están relacionadas a través de la transformación lineal J , que así mismas cambian durante del tiempo que cambia θ . Se piensa que el **Jacobiano** J es lo mejor para una transformación lineal variante en el tiempo ecuación (5.2.2)

Recordando el problema de la cinemática inversa, donde se resuelve para el vector de estados θ dada una posición y una orientación X del resultado final ecuación (5.2.2)

Ahora sí para la más simple de las articulaciones, la función $f(\theta)$ es altamente no lineal, cuando el número de eslabones incrementa esta rápidamente llega a ser mas y más compleja y su inversa pronto llega a ser imposible de realizarse analíticamente. Sin embargo, el problema puede hacerse lineal, localizando la posición de la operación actual e invirtiendo el Jacobiano da:

$$d\theta = J^{-1} (d X) \quad (5.3.1d)$$

y de esa forma iterar hacia la meta, sobre una serie de pasos incrementales. La primera cosa que necesitamos hacer, dado una articulación, es construir el **Jacobiano** [MA92]

5.3.2 Construyendo el Jacobiano.

Si se sabe la **expresión analítica** de la ecuación (5.2.1) entonces siempre se puede evaluar el **Jacobiano** a través de la diferenciación directa [MA92]. Por ejemplo si se sabe que la **expresión analítica** es

$$X_0 = -w_{Pelvis}/2 - (l_{Pierna} * \sin(\theta_3 + \theta_1) + l_{Pierna} * \sin(\theta_1)) * \sin(\theta_2);$$

$$Y_0 = -30 - (l_{Pierna} * \cos(\theta_3 + \theta_1) + l_{Pierna} * \cos(\theta_1));$$

$$Z_0 = (l_{Pierna} * \sin(\theta_3 + \theta_1) + l_{Pierna} * \sin(\theta_1)) * \cos(\theta_2);$$

se construye el **Jacobiano** para una matriz de 3 x 3

$$[J] = \begin{bmatrix} J_a & J_b & J_c \\ J_d & J_e & J_f \\ J_g & J_h & J_i \end{bmatrix}$$

o sea

$$[d(x_0)/d(\theta_1) \quad d(x_0)/d(\theta_2) \quad d(x_0)/d(\theta_3)]$$

$$[d(y_0)/d(\theta_1) \quad d(y_0)/d(\theta_2) \quad d(y_0)/d(\theta_3)]$$

$$[d(z_0)/d(\theta_1) \quad d(z_0)/d(\theta_2) \quad d(z_0)/d(\theta_3)]$$

realizando las respectivas derivadas se obtiene el **Jacobiano J**

5.3.3 Invirtiendo el Jacobiano.

Se resuelve la matriz inversa del **Jacobiano J** usando la **regla de Cramer**:

$$J^{-1} = \text{adj}(J) / \det(J) \quad (5.3.3a)$$

Para calcular el determinante de \mathbf{J} , si $\det = 0$, no existe inversa y se dejan los mismos ángulos $\det = 1$

5.4 Mapear Cambios.

Mapear cambios en los ángulos variables θ a cambios \mathbf{X} en la posición y orientación en el resultado final, ya obtenida la matriz inversa del **Jacobiano** \mathbf{J} , se encuentra una pequeña delta $d\mathbf{x}$, o sea un cambio diferencial:

$$d\mathbf{x} = \begin{bmatrix} [deltax/n] \\ [deltay/n] \\ [deltaz/n] \end{bmatrix} \quad (5.4a)$$

donde :

$$deltax = x-x_0, \text{ deltay} = y-y_0, \text{ deltaz} = z-z_0;$$

y por la ecuación (5.3.1d) se calcula $d\theta$

$$\begin{bmatrix} [d\theta_1] \\ [d\theta_2] \\ [d\theta_3] \end{bmatrix} = \begin{bmatrix} [&] \\ [\mathbf{J}^{-1}] \\ [&] \end{bmatrix} \begin{bmatrix} [deltax/n] \\ [deltay/n] \\ [deltaz/n] \end{bmatrix} \quad (5.4b)$$

entonces se obtienen cambios en los ángulos θ_i al obtener la **nueva** θ_i

$$\text{nueva } \theta_i = \theta_i + d\theta_i \quad (5.4c)$$

Y de esta forma se sigue la iteración hasta llegar a la meta

5.5 Un Paso Iterativo hacia la Meta.

En cada iteración una $d\mathbf{x} = \mathbf{x}_f - \mathbf{x}_i$ puede ser calculada a partir de las posiciones actuales \mathbf{x}_i y de las posiciones del resultado final \mathbf{x}_f . Los ángulos incrementales $d\theta$ pueden ser calculadas usando el jacobiano inverso y $d\mathbf{x}$ antes, para encontrar un nuevo vector $d\theta$ de ángulos de estado. El procedimiento se repite hasta que el resultado final haya alcanzado la meta y \mathbf{J} debe ser recalculada en cada iteración. Note que desde entonces la relación lineal

representada por J es solamente valida para pequeñas perturbaciones en la configuración del manipulador. Esta aproximación de iteración hacia la meta, sobre una serie de pasos incrementales se muestra esquemáticamente en la (figura 5 1a,5 1b) [MA92].

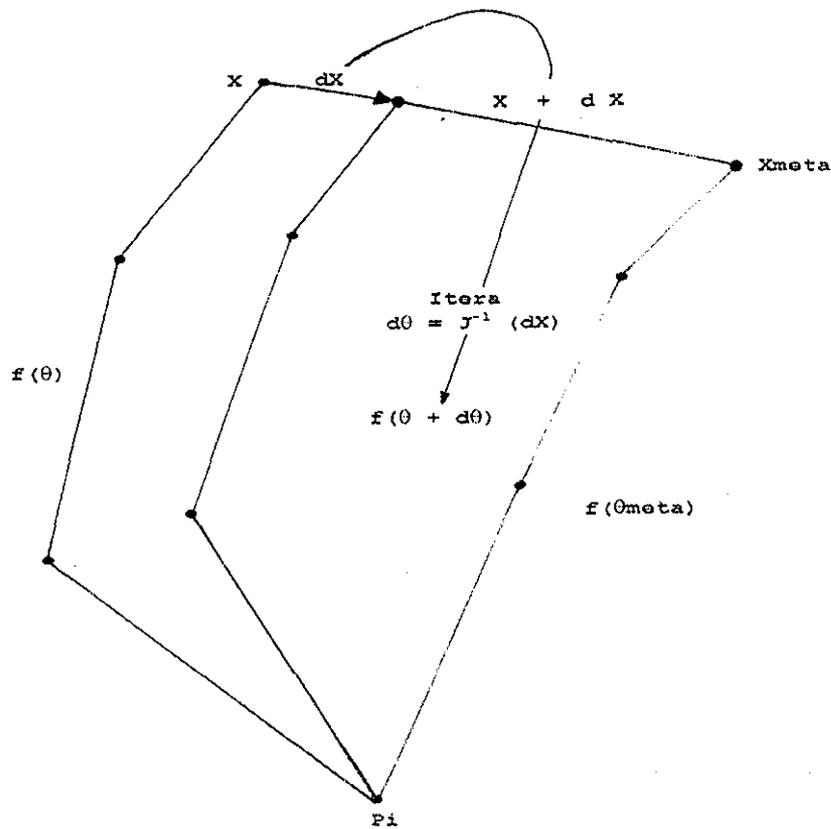


Figura 5.1a Un Paso Iterativo hacia la Meta.

Aquí se está iterando hacia la meta que es la posición final x_{meta} , ahora si nuestro término de velocidad X también contiene la velocidad angular, los cambios diferenciales dX en X contendrán un término incremental del desplazamiento angular $d\theta$, permitiendo así también iterar hacia una orientación de la meta.

$$\text{punto fijo} \rightarrow f(\theta) \rightarrow X$$

$$\text{punto fijo} \rightarrow f(\theta + d\theta) \rightarrow X + dX$$

$$\text{iterativo} = dX \rightarrow d\theta = J^{-1}(dX) \rightarrow f(\theta + d\theta)$$

$$\text{punto fijo} \rightarrow f(\theta \text{ meta}) \rightarrow X_{\text{meta}}$$

Figura 5.1b Un paso iterativo hacia la meta.

5.6 Redundancia.

Un manipulador se considera redundante en la cinemática, cuando este posee mas grados de libertad, de los que son requeridos para especificar una meta o resultado final Para una tarea de posicionamiento simple, la meta es colocar el resultado final (la extremidad de la liga distante de la cadena) en algún **punto** (x, y) Como muestra la (figura 5 2) [MA92], para una meta dada (x, y) , no hay solución única, cada una de las configuraciones mostradas colocaran la extremidad en la posición meta

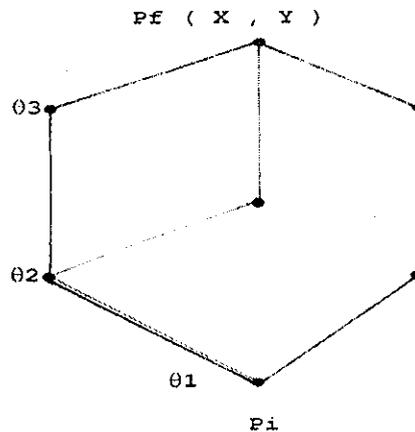


Figura 5.2 Colocar el resultado final en el punto (x, y) .

El manipulador es además redundante para esta tarea de posicionamiento **2D**

En general, posicionando un objeto en el espacio cartesiano requiere la especificación de seis coordenadas: tres para localización y tres para orientación. Además, cualquier manipulador poseyendo más de seis grados de libertad es redundante para la tarea general de posicionamiento en el espacio **3D** y no hay un conjunto único de valores solucionando el problema de cinemática inversa

Para un **manipulador redundante**, la matriz del **Jacobiano** tiene **mas pocos renglones que columnas** y **no puede ser invertido**. En este caso, la ecuación (5.3.c) es **bajo – determinada** y hay un **número infinito de soluciones** de los cuales elegir. Si J^{-1} en (5.3.c) se reemplaza por una inversa generalizada J^* , entonces una solución útil al problema **bajo determinada** puede ser encontrada. Una inversa tal generalizada es la pseudo inversa Moore-Penrose. Esta puede mostrar que la pseudo inversa es optima en el sentido que esta produce soluciones con una norma Euclidiana para casos en que (5.3.c) es **bajo-determinada** ($m > n$) una solución de mínimos cuadrados es obtenida. En la práctica, estas propiedades aseguran que las uniones se mueven tanto como sea posible para emparejar la velocidad del resultado final deseado. Desde entonces un manipulador redundante puede satisfacer una tarea de posicionamiento en cualquier número de caminos, esto es frecuentemente útil al explotar la redundancia en un intento de satisfacer algún criterio secundario, metas secundarias han sido creadas para evitar colisiones con obstáculos [Bai86] explotando rango de ángulos disponibles y aun manteniendo la destreza del manipulador evitando singularidades en cinemática

5.7 Las Singularidades.

El método pseudo inverso provee soluciones útiles a (5.3.c) cuando la matriz del **Jacobiano** J es **rectangular** y además **no invertible**. Pero debemos también considerar el caso donde J no es invertible porque esta es singular. **Una matriz se dice singular cuando dos o más renglones son linealmente dependientes** y un manipulador se dice estar en una configuración singular cuando el **Jacobiano** llega a ser singular. La matriz del **Jacobiano** calculada para esta configuración contendrá ceros en el primer renglón y es además singular y no puede ser invertida. La pseudo inversa puede todavía ser aplicada para obtener una solución cuando J es singular. Sin embargo, cuando un manipulador pasa a través de **una configuración singular hay discontinuidades** en elementos del calculo pseudo inverso debido al cambio en el rango de J en la configuración singular. Además, **si el manipulador se aproxima a esta configuración la pseudo inversa tiende a producir grandes velocidades angulares**. Las técnicas de integración numérica típicamente no manejan bien tales derivadas de puntos. El problema se manifiesta así mismo como una tendencia del manipulador a oscilar desordenadamente alrededor de la configuración singular. Así, mientras el pseudo inverso es capaz de proveer una solución útil en una configuración singular, su principal tiro de regreso es que este no provee una continuidad y una solución estable alrededor de singularidades. Entonces manipuladores de robótica industrial deben ser programados para seguir trayectorias que explícitamente eviten configuraciones singulares y la pseudo inversa realmente no es una opción para ser usada como un algoritmo para la animación en aplicaciones gráficas

La Descomposición del Valor Singular.

Inestabilidades numéricas cerca de configuraciones singulares son un problema principal, que levanta la pregunta de si hay un medio de detectar y corregir el problema. Probablemente la herramienta más útil para analizar la matriz del **Jacobiano** es la descomposición del valor singular (SVD). Mientras el **SVD** **proporciona un medio para detectar mal condicionamiento** en la matriz del **Jacobiano**, este no provee en sí mismo un camino para tratar con el mal condicionamiento. Sin embargo esto es tan útil como una herramienta analítica. Klein y Huang han usado análisis de valor singular para demostrar las propiedades óptimas de la pseudo inversa Moore Penrose. Maciejewski describe un método de actualización incremental del SVD desde una iteración a la próxima que reduce el costo a $O(n^2)$ por iteración, pero esto requiere una implementación cuidadosa para reducir errores acumulados y el costo es todavía bastante alto para impedir su uso.

El rango de una matriz se define como el número más grande de renglones (columnas) linealmente independientes de una matriz. Con los movimientos de la articulación los **Jacobianos** cambiarán con el tiempo y esta variación puede producir discontinuidades súbitas en el rango del **Jacobiano**. Estas singularidades no se causan por cualquier insuficiencia de la formulación pero son inherentes en la transformación de las velocidades del vector de estados a las velocidades cartesianas. Físicamente, las singularidades normalmente ocurren cuando la articulación está totalmente extendida o cuando los ejes de eslabones separados se encuadran. La (figura 5.3) [MA92] muestra una articulación totalmente extendida - en ese instante, los dos cambios en θ_1 y θ_2 producirán cambios en el fin de la articulación, en exactamente la misma dirección perpendicular a los ejes de los eslabones. No hay ningún conjunto de velocidades de vector de estados que darán una velocidad cartesiana hacia o fuera de la base - un grado de libertad ha sido perdido. Ahora los textos en el álgebra lineal le dirán que el determinante de una matriz provee una medida de su rango - si el determinante es igual a cero la matriz ha perdido su rango - lo que podemos usar para calcular las configuraciones singulares de una articulación. Esto es, corresponden a la extensión total o cuando el segundo eslabón regresa completamente plegado en el primero. Hay dos acercamientos alternativos al problema de la singularidad en la extensión total (se dice que es una singularidad de límite de área de trabajo en el idioma de la robótica). El primero simplemente es no permitir la articulación para volverse totalmente extendido - es bastante empujar el fin atrás, hacia la base, por una cantidad muy pequeña y por eso retener ese grado de libertad. El segundo es adoptar un proceso reiterativo diferente dentro de la región de la singularidad. Sims y Zeltzer adoptan la segunda estrategia que usa una unión de la pelota a la base de la articulación para ayudar en su iteración dentro de la región singular. No se acerca al manejo de singularidades interiores que ocurren en las regiones de otra manera a aquéllos en que la articulación es totalmente extendida.

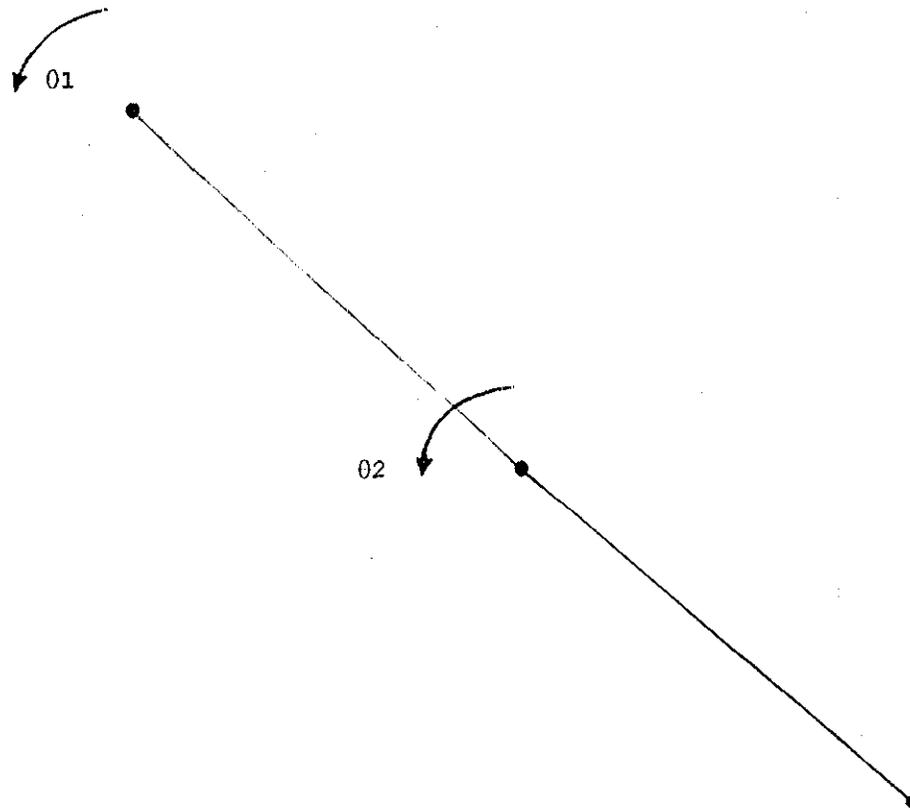


Figura 5.3 Singularidad en Extensión Total

5.8 Condicionamiento.

El condicionamiento ocurre en la **transición entre las configuraciones no singulares y singulares** y los resultados en las **oscilaciones sin control y velocidades del vector de estados inaceptablemente altas**. Simplemente una situación tal se representa en la (figura 5.4) [MA92] que muestra tres configuraciones de una articulación de tres eslabones donde la suma de las longitudes de los últimos dos eslabones es igual al primero. Hay una singularidad así cuando los ejes de los eslabones son todos alineados y el fin se posiciona directamente sobre la base como se muestra en la (figura 5.4(a) y 5.4(b)) [MA92] muestra la articulación en dos configuraciones, ambas cerradas a esta singularidad, donde los primeros eslabones se orientan en las direcciones casi opuestas, uno con el fin ligeramente sobre la base y uno con este ligeramente debajo. Ahora la distancia dX entre estas

posiciones finales en (figura 5 4(b)) [MA92] pueden hacerse arbitrariamente pequeñas pero las distancias cruzaron en el vector de estados entre las dos configuraciones, particularmente el de θ , es comparativamente grande. Como el $d\mathbf{X} \rightarrow 0$, el $d\theta$ no hace el más cercano, su conducta que llega a ser mas y más errática. Determinando el valor requerido así para $d\theta$ dado el $d\mathbf{X}$ no es un problema bien propuesto - se dice que esta mal condicionado.

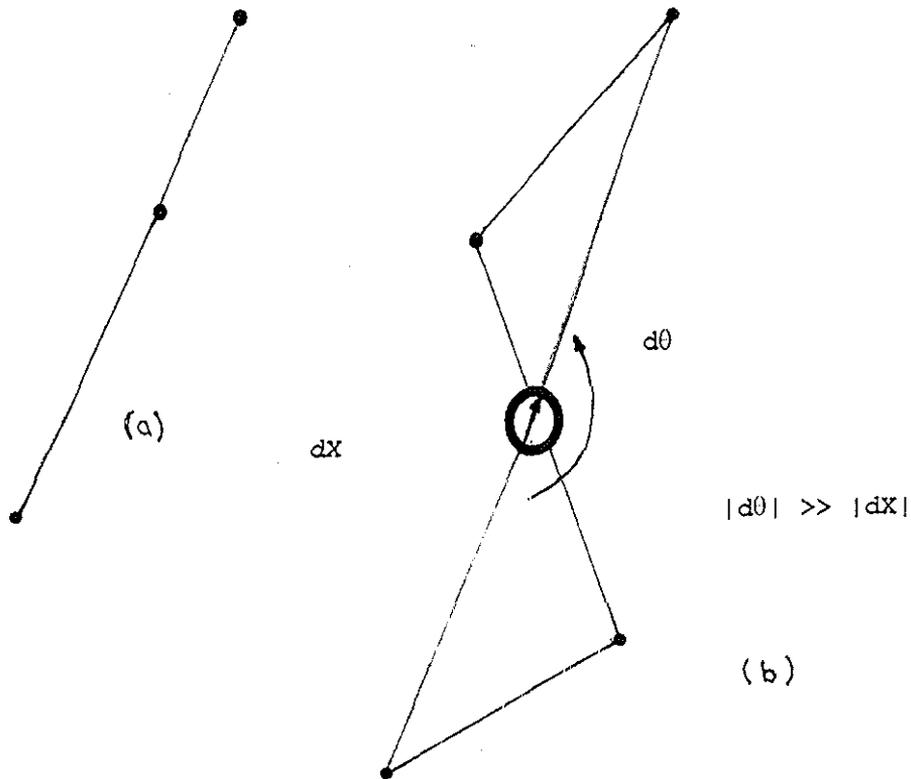


Figura 5.4 Condicionamiento.

La estrategia numérica mas comúnmente usada, llamada mínimos cuadrados amortiguados (damped least square), que ha sido desarrollada para hacer frente a búsquedas mal condicionadas para una solución que minimiza la suma:

$$\|J(d\theta) - d\mathbf{X}\|^2 + \lambda^2 \|d\theta\|^2$$

Donde el segundo término es una medida de las velocidades del vector de estados. La razón es intentar minimizar el error rastreado y las velocidades del vector de estados al mismo tiempo. La minimización del segundo término llamado el factor desanimado λ evita cualquier movimiento espurio. Se usa para poner la importancia relativa del error rastreado contra la norma de las velocidades.

5.9 Métodos basados en Optimización.

Una aproximación diferente resolviendo el problema de cinemática inversa evita el paso total de la matriz inversa. La idea es transformar el problema básico de la ecuación (5.2.2) como un problema de minimización, entonces aplican técnicas de optimización no lineal iterativa estándar para obtener una solución.

Como un ejemplo considere el problema de posicionamiento del resultado final \mathbf{X} , en una meta de posición \mathbf{P} . La distancia desde la posición actual \mathbf{X} a la meta de posición \mathbf{p} sirve como una medida de error:

$$E(\theta) = (\mathbf{p} - \mathbf{x}(\theta))^2 \quad (5.9)$$

Variando el ángulo del vector θ el resultado final se mueve lejos de \mathbf{p} , incrementando la medida del error, o decrementando el error hacia \mathbf{p} . Claramente el intento es encontrar un ángulo de vector θ que minimice la medida del error. El límite sobre el rango de ángulos de movimiento, provee restricciones adicionales, sobre valores de ángulos individuales θ_i . Formalmente se necesita encontrar un vector θ que resuelva el problema

$$\begin{aligned} &\text{Minimiza } E(\theta) \\ &\text{Sujeto a } l_i \leq \theta_i \leq u_i \quad i = 1 \dots n \end{aligned}$$

Donde l_i y u_i son los límites más bajos y más altos, sobre el valor de ángulos variables θ_i . Para este ejemplo, la medida del error E es justamente la fórmula de distancia, la peor aproximación generaliza a metas más complejas para el resultado final. Entonces E puede ser cualquier función arbitraria del ángulo θ del vector. Esta formulación es un problema de optimización de restricciones no lineales clásica, que puede ser resuelta por un número de métodos numéricos estándar.

Gis et Al. examina un número de técnicas de optimización prácticas. Lo menos efectivo de cualquier método particular es determinado usualmente por las características de la función objetivo de las restricciones, en este caso E . Para nuestro ejemplo, una solución para minimizar funciones cuadráticas suaves sujetas a restricciones de desigualdad lineal serían una elección apropiada. Una solución típica convergerá iterativamente desde

un estado inicial hacia una solución de estado, en cada paso perturbando ligeramente las variables de estado y reevaluando la función objetivo para evaluar su progreso. Algunas soluciones deben hacer uso del gradiente de la función objetivo E para sugerir nuevas direcciones en que perturbar el vector de estados. Este puede incrementar el cálculo por iteración, pero paga en mejor proporción de converger hacia una solución.

El Error Numérico

Si el **Jacobiano** no es cuadrado, es decir, si la dimensión de \mathbf{X} no es igual a la dimensión de θ entonces esta es rectangular y también no puede ser invertida (éste casi siempre es el caso). Las técnicas llamadas de pseudo inversa son entonces usadas. Estas técnicas son aproximadas y locales. Por lo tanto el proceso de la iteración, dado un cambio deseado, en el resultado final $d\mathbf{X}$, desde su posición actual, intenta calcular el cambio correspondiente en el vector de estados $d\theta$ por la ecuación (5.3.1.d)

$$d\theta = J^{-1} (d\mathbf{X}) \quad (5.3.1.d)$$

Está sujeto a error. Es más si el $d\mathbf{X}$ ya es demasiado grande entonces, debido a la naturaleza local de la solución, los errores casi ocurrirán ciertamente. A esta clase de errores se les llaman errores de 'rastreo' - la magnitud de estos errores es obtenida de la norma de la diferencia entre el cambio real y el cambio deseado. Esto es:

$$\|J(d\theta) - d\mathbf{X}\|$$

Para ello se describe una estrategia que minimiza estos errores de rastreo. Suponga que el error de rastreo es inaceptable, cuando iteramos sobre el intervalo $[\mathbf{X}, \mathbf{X}_{meta}]$ (es decir, cuando el $d\mathbf{X} = \mathbf{X}_{meta} - \mathbf{X}$), este excede un valor de umbral predefinido. El camino de \mathbf{X} a \mathbf{X}_{meta} se subdivide y la articulación se hace al iterar hacia una posición de meta intermedia y entonces, si esta iteración produce un error aceptable, para iterar desde la configuración intermedia resultante hacia la \mathbf{X}_{meta} . Sin embargo si este error también es demasiado grande, entonces el intervalo apropiado se subdivide y el proceso se repite. Si incrementamos i (siguiendo la ruta más baja) hasta que el error sea suficientemente pequeño, con lo cual la articulación es actualizada (siguiendo la ruta más alta) con la configuración aceptable, es decir, $\mathbf{X} = \mathbf{X} + J(d\theta_i)$ y el proceso empieza uno nuevo.



Capítulo 6.

Animación. El Caminar de la Figura Humana.

La clave para generar la animación del esqueleto con éxito es un análisis cuidadoso de las metas que la cinemática inversa intenta satisfacer, identificando dónde aplicar la cinemática inversa y cómo restringir sus nodos finales [MA92]

6.1 Antecedentes.

La noción de animar el nodo final independientemente de la (posición / orientación) de la meta que intente alcanzar, es sumamente poderosa - en este contexto la meta es normalmente una restricción que la cinemática inversa intenta satisfacer. Así el esqueleto, en un sentido muy real, puede hacer que interactue recíprocamente con su ambiente a través del tiempo. En la animación la restricción no debe ser confundida con el proceso de la iteración para alcanzar el nodo final - este nunca debe ser considerado como parte de la animación. La restricción del nodo final se define y entonces se itera hasta que esta sea satisfecha o cuando esta sea imposible, adoptando con esto la mejor configuración que minimiza la diferencia entre la restricción y el nodo final. En Montreal, SoftImage desarrollo un trabajo de este tipo de animación [MA92].

Sistemas de Animación. La literatura de animación en aplicaciones gráficas rara vez es conducida en forma sintáctica o semántica, en donde las representaciones para mecanismos, logran una técnica de animación o sistema. Como en el caso de algunos sistemas de animación interactiva tales como GRAMPS, TWIXT y BBOP. También como en el caso de paquetes de animación comercial como Alias Wavefront (mecanismo que une un objeto a otro). En este camino, el usuario puede construir jerarquías. Cuando el usuario manipula un objeto, en sus objetos más pequeños no se nota una articulación real. Muchos sistemas de animación no son interactivos y están basados sobre scripts que proveen una jerarquía sólo a través de una interfaz de lenguaje de programación. Ejemplos de tales

sistemas son ANIMA-II, ASAS y MIRA-3D. En esta clase de sistema, el manejo de la jerarquía es difícil dentro del script. Una jerarquía diseñada en este camino es muy limitada, excepto en las manos de un programador talentoso.

6.2 Esqueleto Humano, sus Ejes Coordinados y la Animación.

Hacer un humano sintético es una tarea compleja, incluso modelar el esqueleto. El modelado del esqueleto humano incluye una amplia gama de problemas a resolver, la representación del cuerpo humano, cinemáticas y dinámicas.

Para modelar el cuerpo humano se analizó y seleccionó una representación apropiada. El cuerpo humano consiste de huesos y uniones que forman una estructura jerárquica - el árbol (figura 6.4) [MA92]. Porque el cuerpo humano es muy complejo, se usó el modelo simplificado de un esqueleto. El esqueleto puede representarse como un conjunto de objetos conectados con las uniones. La complejidad del esqueleto es dada por los grados de libertad (DOF) donde un DOF se representa por un eje rotatorio. Así que para conseguir una aproximación más detallada se requieren más de 200 DOF. El esqueleto empleado en este trabajo requirió de 22 DOF aproximadamente, ver figura 6.1.

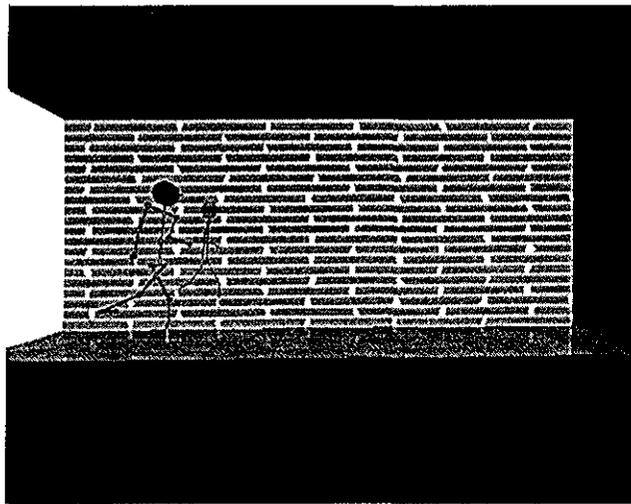
Cinemática.

La cinemática es una ciencia de movimiento sin enfocarse a fuerzas que afectan el movimiento. La cinemática se divide en las cinemáticas directas e inversas. La cinemática directa consigue la posición y orientación de último segmento en una cadena cinemática definiendo los ángulos para cada unión. Por otro lado, la cinemática inversa calcula los ángulos de unión de una cadena cinemática basada en la posición y orientación de último segmento de la cinemática.

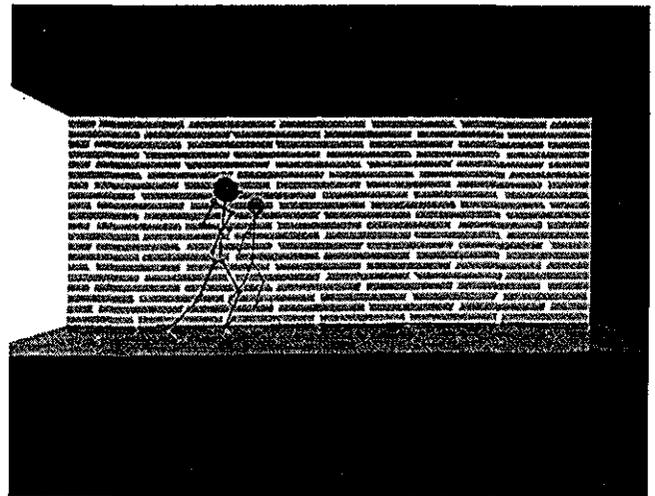
Dinámica. (Este Trabajo no considera la dinámica).

La dinámica considera las leyes físicas, usa varias fuerzas y energía cinética que pueden variar con el tiempo. También la dinámica se divide en dinámica directa e inversa. La dinámica directa explícitamente activa fuerzas y energía cinética (variando a tiempo) a los segmentos del cuerpo. Entonces el movimiento se aproxima a pasos de tiempo discretos, donde los valores de fuerzas y energía cinética son conocidos. La dinámica inversa determina las fuerzas y energía cinética que se necesitan para alcanzar la meta dada.

Ejes Coordinados.



Kn#100



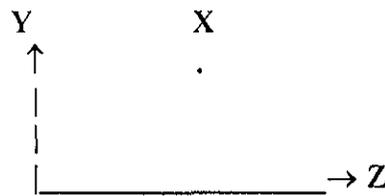
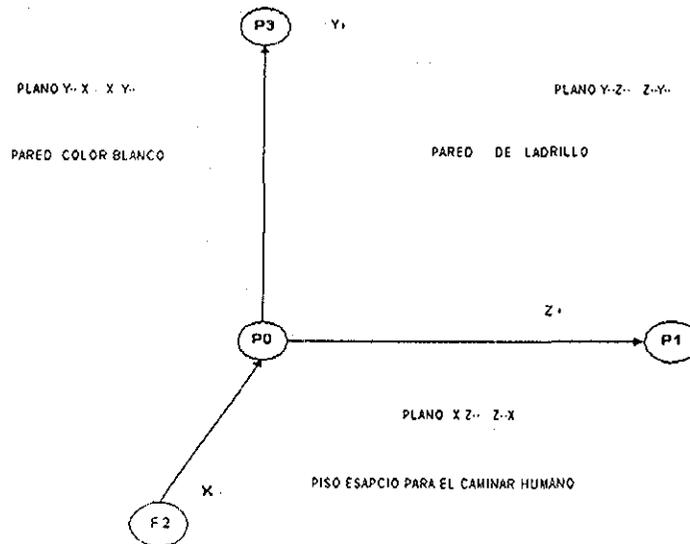
kn#130

Figura 6. 1. Posición Inicial de un paso con la pierna de giro azul kn#100 y sus movimientos en las partes de su cuerpo kn#130.

Los ejes coordenados X, Y, Z y sus planos se muestran en la figura 6.2:

El sistema de coordenadas natural de RenderMan es la mano izquierda en la cual el eje X apunta a la derecha, el eje Y apunta arriba y el eje Z apunta dentro de la pantalla Para nuestro caso como se puede ver el eje X apunta dentro de la pantalla parte blanca de la figura 6.1 y el eje Z apunta a la derecha parte del piso y la pared de ladrillo en la figura 6.1, el cual nos da nuestro espacio para caminar y el espacio para desplazar el cuerpo de la figura articulada En la figura 6.2 se muestra el eje de coordenadas consideradas para esta animación, donde PO es el punto origen de la pelvis o cadera donde a partir de este punto PO y considerando las dimensiones y ángulos de cada parte del cuerpo se define la posición de la figura articulada con sus respectivos ejes coordenados figura 6.1 kn#100, dando en la tabla 6.1 y 6.2 sus posiciones X_0, Y_0, Z_0 para sus brazos y piernas

TESIS CON
FALLA DE ORIGEN



PO (0.0, -30.0,0.0).

Figura 6.2 Ejes coordenados, sus planos y puntos (P0, P1, P2, P3) 3D en el espacio. La Posición PO de los ejes coordenados para colocación de la figura articulada, donde para la animación se rota sobre el eje X y se desplaza el cuerpo sobre el eje Z, como se puede ver en la figura 6.1.

Planos.

A partir de las figuras 6.1, 6.2, (6.5) [SJ99] y de un análisis de trigonometría en sus planos se desarrolla la obtención de las ecuaciones algebraicas que se describe a continuación: En el plano YZ, ZY se encuentra la referencia Pared de ladrillo, donde sus coordenadas son P1(-120 0, -140 0,500 0) eje Z; P0(-120 0, -160 0, -400 0) origen, P3(-120 0,200 0, -400) eje Y. Esta referencia se usa como fondo dentro de la escena para poder proyectar la sombra de la figura articulada. Para el brazo tenemos:

$$\sin(\theta_1 - \theta_3) = Z / l; \quad \cos(\theta_1 - \theta_3) = Y / l;$$

$$Z = l * \sin(\theta_1 - \theta_3); \quad Y = l * \cos(\theta_1 - \theta_3);$$

Para la pierna tenemos: $\sin(\theta_1 + \theta_3) = Z / l$; $\cos(\theta_1 + \theta_3) = Y / l$;

$$Z = l * \sin(\theta_1 + \theta_3); Y = l * \cos(\theta_1 + \theta_3);$$

En el plano XY, YX se encuentra la referencia Pared color Blanco, donde sus coordenadas son P2(0.0,-160.0,-400.0) eje X, P0(-120.0,-160.0,-400.0) origen, P3(-120.0,200.0,-400.0) eje Y. Esta referencia se usa para dar el espacio de profundidad del espacio 3D en la pantalla

Para el brazo: $X = l * \sin(\theta_1 - \theta_3)$; $Y = l * \cos(\theta_1 - \theta_3)$;

Para la pierna: $X = l * \sin(\theta_1 + \theta_3)$; $Y = l * \cos(\theta_1 + \theta_3)$;

Para el plano ZX, XZ se encuentra la referencia Piso espacio para el caminar humano, donde sus coordenadas son P2(0.0, -160.0, -400.0) eje X, P0(-120.0, -160.0, -400.0) origen, P1(-120.0, -140.0, 500.0) eje Z. Esta referencia se usa para indicar el camino que la figura articulada recorrerá al caminar sobre el piso

$$\sin \theta_2 = X / r; \cos \theta_2 = Z / r;$$

Para brazos y piernas: $X = r * \sin \theta_2$; $Z = r * \cos \theta_2$;

Observación : Plano XY= YZ, YX = ZY por lo tanto en estos planos X = Z
De lo anterior, sumando las ecuaciones X, Y y Z de cada ecuación se obtienen las ecuaciones para los brazos y piernas

Ecuaciones del Brazo.

$$X = (l * \sin(\theta_1 - \theta_3) + l * \sin(\theta_1 - \theta_3)) * \sin \theta_2 = (2l * \sin(\theta_1 - \theta_3)) * \sin \theta_2$$

$$Y = l * \cos(\theta_1 - \theta_3) + l * \cos(\theta_1 - \theta_3) = 2l * \cos(\theta_1 - \theta_3)$$

$$Z = (l * \sin(\theta_1 - \theta_3) + l * \sin(\theta_1 - \theta_3)) * \cos \theta_2 = (2l * \sin(\theta_1 - \theta_3)) * \cos \theta_2$$

Ecuaciones de la Pierna.

$$X = (l * \sin(\theta_1 + \theta_3) + l * \sin(\theta_1 + \theta_3)) * \sin \theta_2 = (2l * \sin(\theta_1 + \theta_3)) * \sin \theta_2$$

$$Y = l * \cos(\theta_1 + \theta_3) + l * \cos(\theta_1 + \theta_3) = 2l * \cos(\theta_1 + \theta_3)$$

$$Z = (l * \sin(\theta_1 + \theta_3) + l * \sin(\theta_1 + \theta_3)) * \cos \theta_2 = (2l * \sin(\theta_1 + \theta_3)) * \cos \theta_2$$

Ecuación de los hombros.

$$X = -w\text{Espalda}/2$$

Ecuación de la columna.

$$Y = l_{\text{Columna1}} \cdot \cos\theta_3 + l_{\text{Columna2}} \cdot \cos\theta_3$$

$$Z = l_{\text{Columna1}} \cdot \sin\theta_3 + l_{\text{Columna2}} \cdot \sin\theta_3$$

Ecuación de las caderas.

$$X = -w_{\text{Pelvis}}/2$$

Ecuación para conservar el origen.

$$Y = -300$$

De esta forma se obtienen las ecuaciones algebraicas (6.4.1) y (6.4.2) de la figura articulada.

Análisis del Movimiento de la Figura 6.1.

Partiendo de la posición inicial se dan ángulos en las partes del cuerpo del modelado humano representado por la figura articulada (esqueleto), donde puede observar la pierna izquierda que gira y el brazo derecho colocados hacia atrás, en forma opuesta la pierna derecha apoyada en el piso y el brazo izquierdo, produciendo la configuración o pose de la figura 6.1 donde sus partes del cuerpo (brazos, pierna, pies) y el desplazamiento del cuerpo, van a tener un comportamiento tal en su movimiento para llegar a la pose final de la figura 6.1. Para ello se tiene que identificar las posiciones iniciales y las posiciones meta o de resultado final, en algunas de las partes del cuerpo como son:

Brazo Izquierdo (inicial, meta), Brazo Derecho (inicial, Meta), Pierna Izquierda (inicial, meta), Pierna Derecha (inicial, meta), Pie Izquierdo (inicial, meta), Pie Derecho (inicial, meta) Estas son las partes del cuerpo de la figura articulada, que usan el método de la cinemática en combinación con el desplazamiento del cuerpo, de los movimientos laterales de los hombros y caderas que darán el efecto de movimiento esperado para la animación del caminar humano y al unir cada una de las poses generadas en cada paso en un programa llamado Paint Shop Pro 7, que coloca cada una de las poses en secuencia y genera la animación que puede ser almacenada en un archivo con extensión avi por ejemplo animación.avi

Términos para poder describir el movimiento

Los términos usados son los nombres que se asignan a los brazos y piernas del cuerpo humano y facilitar su descripción durante el movimiento

Pierna derecha (color rojo) [cadera, rodilla], lo llamo PrD [c2, r2, t2]

Pie derecho (color rojo), lo llamo Trd

Pierna izquierda (color azul) [cadera, rodilla], lo llamo Pal [c1, r1, t1]

Pie izquierdo (color azul), lo llamo Tri

Movimientos en sentido horario que ocurren al mismo tiempo

True Brd[b2 -= 1 0], PrD[c2 -=1 6], avance += 1 9, pelvis += 1 6 parte derecha, todo el cuerpo, pelvis+ (hombros-, caderas+).
Brazo izquierdo (color azul) [brazo, codo], lo llamo BaI [b1]

Movimientos en sentido antihorario que ocurren al mismo tiempo

False BaI[b1 -= 1 0], PaI[c1 -=1 6], avance += 1 9, pelvis -= 1 6 parte izquierda, todo el cuerpo, pelvis- (hombros +, caderas -)
Brazo derecho (color rojo) [brazo, codo], lo llamo Brd, [b2]

Movimiento hombros (- gira en sentido negativo sobre y, + gira en sentido positivo sobre y)

Pelvis += 1 6

Movimiento caderas, pelvis (en forma inversa a los hombros)

Pelvis -= 1 6

Cinemática inversa aplicada a brazos o piernas

ki()

Descripción del movimiento en un ciclo de caminar humano.

Las metas a considerar para el movimiento del caminar humano son:

PaI giro, atrás	→	PaI apoyo, adelante	BaI adelante	→	BaI atrás
PrD apoyo, adelante	→	PrD giro, atrás	BrD atrás	→	BrD adelante

Las posiciones de inicio son:

PaI giro, atrás	BaI adelante
PrD apoyo, adelante	BrD atrás

Primer paso del caminar humano generado para la figura articulada.

Se cuenta con las posiciones iniciales anteriores se realizan los movimientos en sentido antihorario de Prd con desplazamiento de ángulo y BaI aplicando ki(BaI), después los movimientos en sentido horario de PaI y Brd aplicando ki(Brd) y ki(PaI). Este proceso usa

interpolación lineal y se repite hasta llegar a la meta de cada pierna y brazo para generar el paso:

Pal giro, atrás → Pal apoyo, adelante Bal adelante → Bal atrás
 PrD apoyo, adelante → PrD levantamiento, atrás Brd atrás → Brd adelante

De esta manera crea un frame o pose que se nombra kn#100.tif que significa pose número cien generado por el algoritmo de animación ver anexo B y C, empleando el método de cinemática.

Segundo paso del caminar humano generado para la figura articulada.

Es lo mismo pero en sentido opuesto.

El ciclo de caminar se repite al iniciar con el primer paso y terminar con el segundo paso

La animación del caminar se forma de dos formas.

La primera Ejecutar el programa Ky nombre de archivo, genera nombre de archivo

Ejecutar: rgl -sync 30 nombre de archivo genera escena de animación.

La segunda. Ejecutar el programa ky rendrib genera los archivos kn//numero.tif (número corresponde al frame o pose para poder generar la escena del movimiento)

Ejecutar Jasc paint shop Pro añadiendo el número de archivos que componen la animación, generando así la animación en un archivo, con un nombre de archivo por ejemplo animacion1.avi con extensión avi

Aplicando Cinemática en la figura 6.1.

Cinemática Directa. En Brazos, se dan los ángulos dados en la tabla 6.1, se obtiene X0, Y0, Z0

a) Brazo Izquierdo

$$\begin{aligned} X0 &= -wEspalda/2 - 2 * lArm * \sin(-12.03) * \sin(0.0) = -20.0 \\ Y0 &= -30 + lColumna1 * \cos(72.19) + lColumna2 * \cos(72.19) - \\ &\quad (2 * lArm * \cos((-12.03) - 72.19)) = -19.87 \\ Z0 &= lColumna1 * \sin(72.19) + lColumna2 * \sin(72.19) \\ &\quad + (2 * lArm * \sin((-12.03) - 72.19)) * \cos(0.0) = 6.23 \end{aligned}$$

b) Brazo Derecho

$$\begin{aligned} X0 &= -wEspalda/2 - 2 * lArm * \sin(-11.22) * \sin(0.0) = -20.0 \\ Y0 &= -30 + lColumna1 * \cos(18.69) + lColumna2 * \cos(18.69) - \\ &\quad (2 * lArm * \cos((-11.22) - 18.69)) = -21.86 \\ Z0 &= lColumna1 * \sin(18.69) + lColumna2 * \sin(18.69) \\ &\quad + (2 * lArm * \sin((-11.22) - 18.69)) * \cos(0.0) = 8.96 \end{aligned}$$

Cinemática Directa. En Piernas, se dan los ángulos dados en la tabla 6.1, se obtiene X_0, Y_0, Z_0 .

a) Pierna Izquierda

$$X_0 = -w_{Pelvis}/2 - (l_{Pierna} * \sin((-79.96) + (-0.41))) + l_{Pierna} * \sin((-0.41)) * \sin(0.0) = -10.0$$

$$Y_0 = -30 - (l_{Pierna} * \cos((-79.96) + (-0.41))) + l_{Pierna} * \cos((-0.41)) = -67.26$$

$$Z_0 = (l_{Pierna} * \sin((-79.96) + (-0.41))) + l_{Pierna} * \sin(\theta_1) * \cos(\theta_2) = -44.56$$

Un resumen de la cinemática directa de sus ángulos y resultados X_0, Y_0, Z_0 en la siguiente tabla 6.1 :

	θ_1	θ_2	θ_3	X_0	Y_0	Z_0		θ_1	θ_2	θ_3	X_0	Y_0	Z_0
BI	-12.03	0.0	72.19	-20.0	-19.87	6.23	PD						
BD	-11.22	0.0	18.69	-20.0	-21.86	8.96	PI	-0.41	0.0	-79.96	-10.0	-67.26	-44.56

Tabla 6.1 Posición y orientación inicial en brazos y pierna izquierda de la figura articulada ver figura 6.1 kn#100.

Habiendo aplicado la cinemática directa, generamos las posiciones iniciales X_0, Y_0, Z_0 , ahora requerimos encontrar sus posiciones finales o sea alcanzar las posiciones de la figura 6.1 kn#130. Para ello aplicamos cinemática inversa

Cinemática inversa, en Brazos dando la posición final X, Y, Z , obtiene los ángulos dados en la tabla 6.2

a) Brazo Izquierdo. Con este fin llamamos a la rutina **AnimaArmIzquierdo ()** que calcula la cinemática inversa cuyo algoritmo esta en el anexo B y C, donde sus parámetros (son la posición final $d1=X, e1 = Y, f1 = Z$ y h es el modelo del cuerpo humano de la figura articulada que contiene los ángulos que dan la posición inicial) son los datos de entrada requeridos, para regresar en h los ángulos correspondientes a la posición final del brazo izquierdo

AnimaArmIzquierdo(d1, e1, f1, h);

b) **Brazo Derecho.** Con este fin llamamos a la rutina **AnimaArmDerecho ()** que calcula la cinemática inversa cuyo algoritmo esta en el anexo B y C, donde sus parámetros (son la posición final $d2=X$, $e2 = Y$, $f2 = Z$ y h es el modelo del cuerpo humano de la figura articulada que contiene los ángulos que dan la posición inicial) son los datos de entrada requeridos, para regresar en h los ángulos correspondientes a la posición final del brazo derecho.

AnimaArmDerecho(d2, e2, f2, h);

Cinemática inversa, en Piernas dando la posición final X, Y, Z , obtiene los ángulos, ver tabla 6 2.

a) **Pierna Izquierda.** Con este fin llamamos a la rutina **AnimaPieIzquierdo ()** que calcula la cinemática inversa cuyo algoritmo esta en el anexo B y C, donde sus parámetros (son la posición final $a1=X$, $b1 = Y$, $c1 = Z$ y h es el modelo del cuerpo humano de la figura articulada que contiene los ángulos que dan la posición inicial) son los datos de entrada requeridos, para regresar en h los ángulos correspondientes a la posición final de la pierna izquierda.

AnimaPieIzquierdo(a1, b1, c1, h)

b) **Pierna Derecha.** Con este fin llamamos a la rutina **AnimaPieDerecho ()** que calcula la cinemática inversa cuyo algoritmo esta en el anexo B y C , donde sus parámetros (son la posición final $a2 = X$, $b2 = Y$, $c2 = Z$ y h es el modelo del cuerpo humano de la figura articulada que contiene los ángulos que dan la posición inicial) son los datos de entrada requeridos, para regresar en h los ángulos correspondientes a la posición final de la pierna derecha.

AnimaPieDerecho(a2, b2, c2, h);

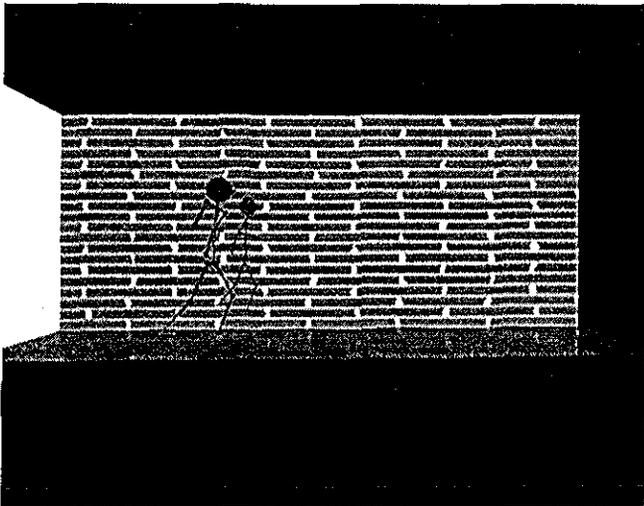
En la tabla 6 2 se resumen los resultados de los ángulos obtenidos dadas las posiciones finales para los brazos y piernas Como nota aclaratoria si la meta o posición final es muy lejana se interpola y se va acercando con desplazamientos pequeños hasta llegar o aproximarse en la iteración a la solución cerrada o buscada

	θ_1	θ_2	θ_3	X	Y	Z		θ_1	θ_2	θ_3	X	Y	Z
BI	-26.44	0.0	12.84	-20.0	-16.69	-15.61	PD						
BD	-20.19	0.0	104.84	-20.0	-18.06	10.70	PI	33.15	0.0	-49.18	-10.0	-83.62	5.37

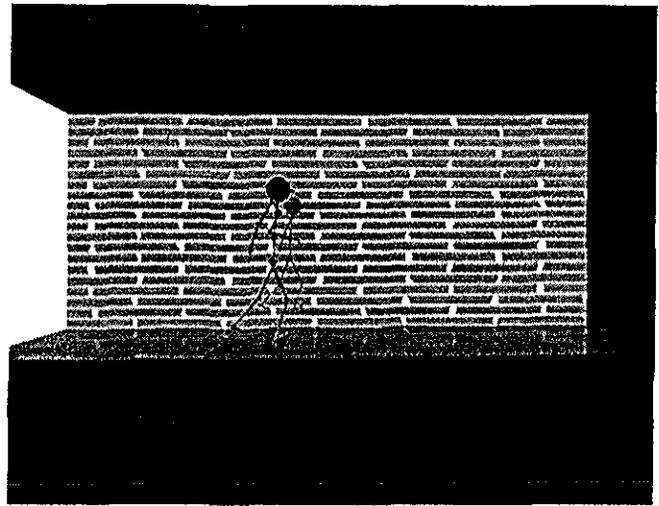
Tabla 6.2. Ángulos encontrados de las posiciones finales de los brazos y piernas.

En la figura 6.1 Kn#100 se da la posición inicial del cuerpo, si observa el sentido que seguirán cada una de las partes del cuerpo, hasta alcanzar las metas de la figura 6.1 kn#130, encontrará, las rotaciones opuestas en sus brazos y piernas, que estas alcanzarán al emplear cinemática directa y cinemática inversa para este movimiento

Siguiendo con el análisis de sus pasos un nuevo paso del caminado en su fase de apoyo y giro se presentan en la siguiente figura 6.3.



Kn#131



kn#160

Figura 6.3 Paso siguiente kn#131 con la otra pierna de giro kn#160.

La explicación del proceso al aplicar cinemática en brazos y piernas se repite para todos los siguientes pasos, a continuación se muestra en la tabla 6.3 sus respectivos resultados.

	θ_1	θ_2	θ_3	X 0	Y0	Z0		θ_1	θ_2	θ_3	X 0	Y0	Z0
BI	-21.10	0.0	20.42	-20.0	-17.09	-15.32	PD	7.89	0.0	-62.99	-10.0	-80.33	-30.96
BD	-17.18	0.0	94.10	-20.0	-17.44	11.57	PI						

	θ_1	θ_2	θ_3	X	Y	Z		θ_1	θ_2	θ_3	X	Y	Z
BI	-22.46	0.0	99.83	-20.0	-15.75	10.43	PD	29.17	0.0	-41.84	-10.0	-85.44	5.30
BD	23.88	0.0	9.04	-20.0	-18.75	-14.81	PI						

Tabla 6.3. Resultados de la figura 6.3 de su posición inicial a su posición final.

TESIS CON
FALLA DE ORIGEN

La figura 6.3 kn#131 da la posición inicial del cuerpo, y si se observa seguirá el movimiento que tendrán cada una de las partes del cuerpo hasta alcanzar las metas de la figura 6.3 kn#160, por ejemplo el brazo azul de la figura 6.3 kn#131 alcanza la posición como se ve en la figura 6.3 kn#160, al emplear cinemática para este movimiento, de manera similar se aplica a las otras partes del cuerpo

Longitud del Paso.

Se considero que un paso equivale a 50 unidades aproximadamente obteniéndose a partir de las posiciones iniciales de Z0 ver tabla 6.1 y 6.2, es decir $44.56 + 5.37 = 49.93 \cong 50$ unidades. De esta forma podemos intentar medir los pasos en cm para estas unidades por ejemplo:

Si 50 unidades = 5cm y 50 unidades equivalen a 1 paso, en cuantos pasos habré recorrido cuanta distancia. Para nuestro algoritmo elegimos 9 iteraciones o pasos, en este caso habremos recorrido $\#pasos * 5cm /pasos = 9pasos * 5cm/pasos = 45cm$ distancia recorrida en 9 pasos. Por lo tanto para este ejemplo la longitud del paso es igual a 5cm, lo que en la realidad del caminado estándar varia de persona a persona al incluirse otros factores tales como peso, estatura, edad

6.3 Parámetros del Movimiento.

Los parámetros que generan movimiento en una figura articulada son los ángulos $\theta_i(\theta_1, \theta_2, \theta_3)$ y sus posiciones $X_i(x, y, z)$. Qué son la clave para lograr movimiento en alguna parte del esqueleto. El asignar valores a los ángulos θ_i en cada parte del esqueleto nos proporciona una configuración o pose inicial al aplicar la cinemática directa. Generando así las posiciones X_i del esqueleto. Los ángulos iniciales considerados son:

Ángulos:

Cuello.

$h \rightarrow xRotCuello = 0.0;$

$h \rightarrow yRotCuello = 0.0;$

Torso, Espalda, Pelvis, Caderas, Hombros

$h \rightarrow xRotEspalda = 0.0;$

$h \rightarrow yRotEspalda = 0.0;$

$h \rightarrow zRotEspalda = 0.0;$

$h \rightarrow RotPelvis = -3.0;$

Brazos.

$h \rightarrow xRotBrazo1 = -14.61; //$ Brazo Izquierdo

$h \rightarrow yRotBrazo1 = 0.0;$

$h \rightarrow zRotBrazo1 = 0.0;$

$h \rightarrow xRotCodo1 = 89.66; //$ Codo Izquierdo

```

h->xRotBrazo2    = -13.93; // Brazo Derecho
h->yRotBrazo2    = 0.0;
h->zRotBrazo2    = 0.0;
h->xRotCodo2     = 10.60;  // Codo Derecho
    
```

Piernas

```

h->xRotCadera1   = -33.0;  // Pierna Izquierda
h->yRotCadera1   = 0.0;
h->zRotCadera1   = 0.0;
    
```

```

h->xRotRodilla1  = -29.0;  // Rodilla Izquierda
h->xRotTobillo1  = 0.0;
h->xRotPuntaPie  = 0.0;
    
```

```

h->xRotCadera2   = 33.0;   // Pierna Derecha
h->yRotCadera2   = 0.0;
h->zRotCadera2   = 0.0;
h->xRotRodilla2  = -29.0;  // Rodilla Derecha
h->xRotTobillo2  = 0.0;
h->xRotPuntaPie  = 0.0;
    
```

Con los ángulos anteriores de brazos, piernas y pies se realiza el cálculo de cinemática directa que dan su posición y orientación de ellos y al considerar los ángulos para todo el cuerpo se obtiene su configuración o pose inicial ver figura 6.1

6.4 Cinemática Directa Aplicada a los Esqueletos.

Las cinemáticas directas explícitamente ponen las rotaciones en las uniones (las transformaciones locales). Entonces la posición y orientación del último segmento en una cadena de la cinemática son dadas por la configuración de la cadena. La transformación global (transformando el cuerpo entero) se da transformando la raíz de la estructura del cuerpo. Allí ocurre una transformación geométrica básica - la rotación.

De acuerdo a la ecuación 5.2.1 la cinemática directa dice: dado los ángulos de las uniones $\theta_i(\theta_1, \theta_2, \theta_3)$ se obtiene la posición y orientación $\mathbf{x}_i(x, y, z)$, en este trabajo se usa solo en brazos, piernas y pies, por ello aplicamos esta técnica a la expresión analítica de la ecuación (6.4.1) obtenida del análisis del cuerpo humano usando trigonometría. Se tratan los ángulos para brazos y piernas aplicados a la ecuación (6.4.1), mostrando sus resultados a continuación:

$$X0 = -wEspalda/2 - 2 * lArm * \sin(\theta1) * \sin(\theta2);$$

$$Y0 = -30 + lColumna1 * \cos(\theta3) + lColumna2 * \cos(\theta3) - (2 * lArm * \cos(\theta1 - \theta3)); \quad (6.4.1)$$

$$Z0 = lColumna1 * \sin(\theta3) + lColumna2 * \sin(\theta3) + (2 * lArm * \sin(\theta1 - \theta3)) * \cos(\theta2);$$

Brazo Izquierdo (Color Azul).

$$\begin{aligned} \theta1 &= \text{hum} \rightarrow x\text{RotBrazo1}/57.295f; \\ \theta2 &= \text{hum} \rightarrow y\text{RotBrazo1}/57.295f; \\ \theta3 &= \text{hum} \rightarrow x\text{RotCodo1}/57.295f; \end{aligned}$$

y los valores para las variables de la estructura h que usa esta ecuación son:

$$\begin{aligned} wEspalda &= 40, lArm = 40, lColumna = 40. \\ lColumna1 &= lColumna/2 = 20. \\ lColumna2 &= lColumna/2 + 5 = 25. \end{aligned}$$

$\theta1 = -14.61$	$\theta2 = 0.0$	$\theta3 = 89.66$
$X0 = -20.0$	$Y0 = -19.87$	$Z0 = 6.23$

Brazo Derecho (Color Rojo).

$$\begin{aligned} \theta1 &= \text{hum} \rightarrow x\text{RotBrazo2}/57.295f; \\ \theta2 &= \text{hum} \rightarrow y\text{RotBrazo2}/57.295f; \\ \theta3 &= \text{hum} \rightarrow x\text{RotCodo2}/57.295f; \end{aligned}$$

$\theta1 = -13.93$	$\theta2 = 0.0$	$\theta3 = 10.60$
$X0 = -20.0$	$Y0 = -22.15$	$Z0 = -8.32$

Cinemática Directa aplicada en piernas

$$X0 = -wPelvis/2 - (lPierna * \sin(\theta3 + \theta1) + lPierna * \sin(\theta1)) * \sin(\theta2);$$

$$Y0 = -30 - (lPierna * \cos(\theta3 + \theta1) + lPierna * \cos(\theta1)); \quad (6.4.2)$$

$$Z0 = (lPierna * \sin(\theta3 + \theta1) + lPierna * \sin(\theta1)) * \cos(\theta2);$$

Pierna Izquierda (Azul).

$$\theta_1 = \text{hum} \rightarrow x\text{RotCadera1}/57.295f;$$

$$\theta_2 = \text{hum} \rightarrow y\text{RotCadera1}/57.295f;$$

$$\theta_3 = \text{hum} \rightarrow x\text{RotRodilla1}/57.295f;$$

y los valores para las variables de esta ecuación son:

$$w\text{Pelvis} = 20, l\text{Pierna} = 60.$$

$\theta_1 = -33.0$	$\theta_2 = 0.0$	$\theta_3 = -29.0$
$X_0 = -10.0$	$Y_0 = -69.24$	$Z_0 = -42.82$

Pierna Derecha (Color Rojo).

$$\theta_1 = \text{hum} \rightarrow x\text{RotCadera2}/57.295f;$$

$$\theta_2 = \text{hum} \rightarrow y\text{RotCadera2}/57.295f;$$

$$\theta_3 = \text{hum} \rightarrow x\text{RotRodilla2}/57.295f;$$

$\theta_1 = 33.0$	$\theta_2 = 0.0$	$\theta_3 = -29.0$
$X_0 = -10.0$	$Y_0 = -85.08$	$Z_0 = 18.43$

6.5 Cinemática Inversa Aplicada a los Esqueletos.

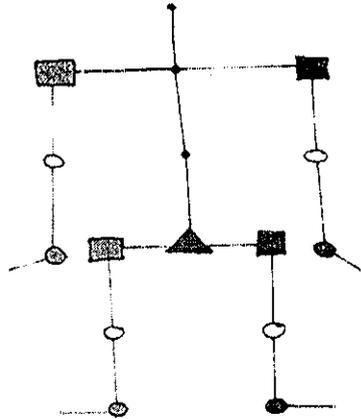


Figura 6.4 Esqueleto Humano.

El método de cinemática inversa es capaz de unir una estructura de esqueleto arbitraria. Si esta contiene una red jerárquica de nodos, uno de estos es la raíz, el resto son descendientes de esta raíz, donde cualquier nodo puede tener cualquier número de hijos. Descendiendo por la raíz de la jerarquía se alcanza cualquier nodo y recíprocamente subiendo por cualquier nodo de la jerarquía siempre terminamos en la raíz. Entonces se aplica la cinemática inversa entre cualquiera dos nodos del esqueleto, la única regla a ser observada es que el nodo que corresponde al resultado final, es el más bajo en la jerarquía y se le llama el nodo final y el nodo base es el nodo que corresponde a la base o parte fija. A los nodos que están entre el nodo final y el nodo base se les llama nodos vacíos. El propósito de la cinemática inversa es especificar la posición y orientación de todos los nodos entre el nodo final y el nodo base. La (figura 6.4) [MA92] muestra un arreglo obvio del esqueleto humano. La raíz denotada por el triángulo negro, está en la pelvis, los nodos base, denotados por los cuadros negros, están en las caderas y hombros, los nodos finales, denotados por los círculos negros, están en las manos y pies y los nodos vacíos, denotados por los círculos perfilados, están en los codos y rodillas.

El nodo final es iterado hasta que este sea satisfecho o cuando esta es imposible, adopta la mejor configuración posible que minimiza la diferencia entre la restricción (ángulos establecidos) y el nodo final.

La cinemática inversa usa una cadena de la cinemática. La cadena de la cinemática es una sucesión de segmentos y uniones. La primera unión es una base que no se mueve y la

última unión que se mueve para dar un resultado final con estos segmentos. La posición y orientación de cada unión está cambiando basada sobre el movimiento del resultado final.

La cinemática inversa ofrece tres maneras de cómo conseguir la solución. Los posibles métodos son algebraicos, geométricos y iterativos. Los métodos algebraicos y geométricos proporcionan una solución exacta (y si allí existen más soluciones entonces los métodos darán todas las posibles soluciones) si la solución existe. Pero algunos problemas de la cinemática no tienen ninguna solución si el blanco para el resultado final es inaccesible. Los métodos iterativos ofrecen una solución general de cinemáticas inversas. Su desventaja es que ellos convergen a sólo una solución aun cuando allí existan más o ellos encuentren una solución más cerrada si no existe.

Los métodos geométricos usan el conocimiento de geometría del manipulador. Este método tiene una desventaja que no pueden usarse soluciones para un manipulador con geometría diferente.

Para resolver las cinemáticas inversas con los métodos algebraicos se tienen que resolver las ecuaciones para los N grados de libertad.

La solución iterativa está basada sobre la inversión de la matriz o sobre cualquier forma de optimización. La inversión de la matriz es un proceso complejo que no sólo su cálculo es difícil sino que pueden ocurrir problemas que vienen de su inestabilidad numérica. Los métodos de optimización desvían el problema de inversión de la matriz. Estos métodos minimizan el error en la cadena de la cinemática. La aproximación iterativa está basada sobre soluciones cerradas continuas para cada unión en una cadena. En general estos métodos son más inexactos y ellos sólo convergen a una solución. Pero estos métodos son convenientes y bastante favorables para la simulación de la cinemática.

En el anexo B y C se muestra el algoritmo de la cinemática inversa que es empleado en los brazos y piernas cada vez que se desplaza la diferencial dx para alcanzar la meta. Esta meta varía para cada brazo ya sea izquierdo o derecho, también para cada pierna izquierda o derecha. Estas metas se muestran a continuación:

Ciclo impar $J = 1$

Brazo	X	Y	Z
Izquierdo	$D1 = -20$	$e1 = -19$	$f1 = -14$
Derecho	$D2 = -20$	$e2 = -21$	$F2 = 11$

Pierna	X	Y	Z
Izquierda	$A1 = -10$	$B1 = -86$	$C1 = 21$
Derecha	$A2 = -10$	$B2 = -86$	$C2 = -21$

Ciclo par $J = 0$

Brazo	X	Y	Z
Izquierdo	D1 = -20	e1 = -19	f1 = 11
Derecho	D2 = -20	e2 = -21	F2 = -14

Pierna	X	Y	Z
Izquierda	A1 = -10	B1 = -86	C1 = -21
Derecha	A2 = -10	B2 = -86	C2 = 21

Ahora ya con las metas anteriores se tiene la posición meta X_i a alcanzar por cada brazo y pierna quedando solo por aplicar el algoritmo de cinemática inversa (ver anexo B y C) para encontrar los ángulos en cada unión para cada brazo y pierna y de esta forma generar las poses para la unión

5.6 Cinemática Inversa en Brazos, Piernas y Pies.

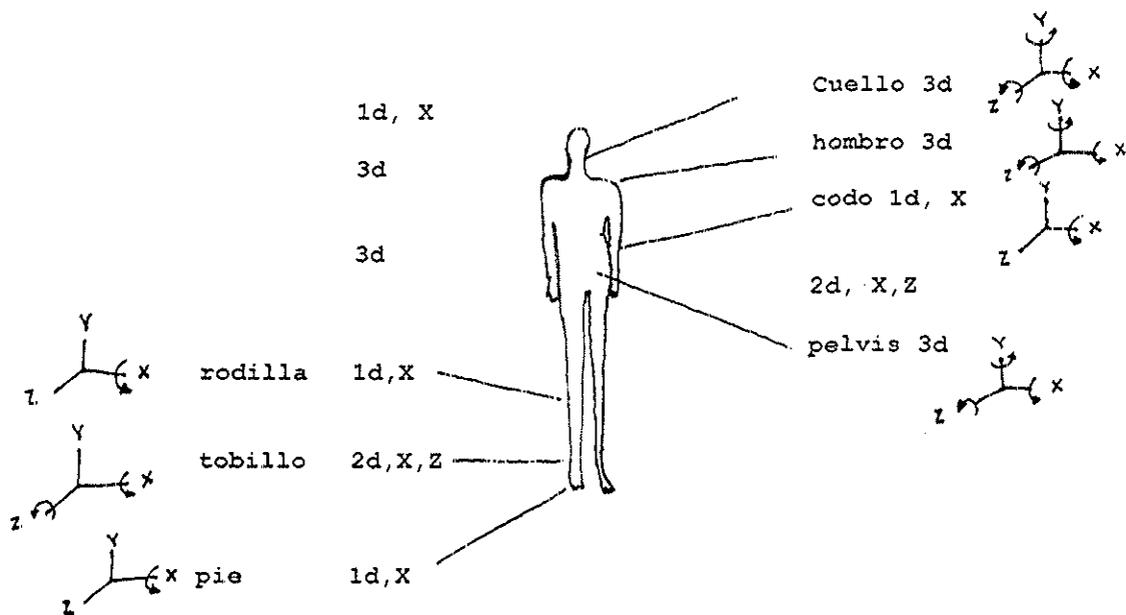


Figura 6.5 Los Grados de Libertad del Cuerpo Humano.

En la (figura 6 5) [SJ99] se muestran los grados de libertad de cada parte del cuerpo [SJ99], el modelo usa solo los siguientes de esta figura, hombros(shoulders), caderas(hip), pelvis, codos (elbows), conejos(bíceps), antebrazos (forearms), muslos (thighs), rodillas (knees), pie(foot) y se consideran las rotaciones en sus respectivos ejes

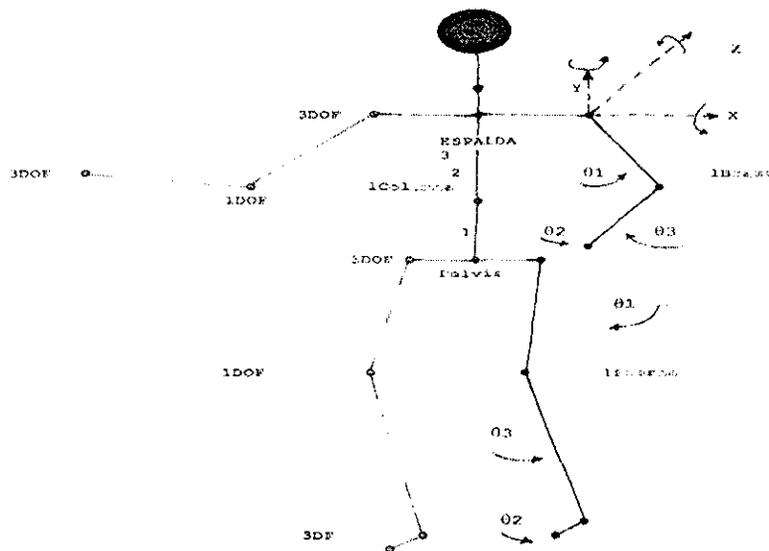


Figura 6.5 La Figura Articulada y sus grados de libertad.

6.6.1 Expresión Analítica del Brazo.

Para describir la expresión analítica del **Brazo Izquierdo** se requiere de un análisis del movimiento de estos brazos ver capítulo 3

Brazo Izquierdo.

A partir de la (figura 6 5) [SJ99] y usando trigonometría se obtienen las expresiones analíticas de la ecuación (6.6 1). Donde los valores para los ángulos ($\theta_1, \theta_2, \theta_3$) del brazo izquierdo son:

- θ_1 = ángulo Brazo Izquierdo en X
- θ_2 = ángulo Brazo Izquierdo en Y
- θ_3 = ángulo Codo Izquierdo en X

$$\begin{aligned}\theta_1 &= \text{hum} \rightarrow x\text{RotBrazo1}/57.295f; \\ \theta_2 &= \text{hum} \rightarrow y\text{RotBrazo1}/57.295f; \\ \theta_3 &= \text{hum} \rightarrow x\text{RotCodo1}/57.295f;\end{aligned}$$

y los valores para las variables de esta ecuación son:

$$\begin{aligned}w\text{Espalda} &= 40, l\text{Arm} = 40, l\text{Columna} = 40. \\ l\text{Columna1} &= l\text{Columna}/2 = 20. \\ l\text{Columna2} &= l\text{Columna}/2 + 5 = 25.\end{aligned}$$

Así las expresiones analíticas obtenidas dan la siguiente ecuación:

$$X_0 = -w\text{Espalda}/2 - 2 * l\text{Arm} * \sin(\theta_1) * \sin(\theta_2);$$

$$Y_0 = -30 + l\text{Columna1} * \cos(\theta_3) + l\text{Columna2} * \cos(\theta_3) - (2 * l\text{Arm} * \cos(\theta_1 - \theta_3)); \quad (6.6.1)$$

$$Z_0 = l\text{Columna1} * \sin(\theta_3) + l\text{Columna2} * \sin(\theta_3) + (2 * l\text{Arm} * \sin(\theta_1 - \theta_3)) * \cos(\theta_2);$$

Y la construcción del Jacobiano [J] se forma de la siguiente manera:

$$\begin{bmatrix} J_a & J_b & J_c \\ J_d & J_d & J_f \\ J_g & J_h & J_i \end{bmatrix} = \begin{bmatrix} d(x_0)/d(\theta_1) & d(x_0)/d(\theta_2) & d(x_0)/d(\theta_3) \\ d(y_0)/d(\theta_1) & d(y_0)/d(\theta_2) & d(y_0)/d(\theta_3) \\ d(z_0)/d(\theta_1) & d(z_0)/d(\theta_2) & d(z_0)/d(\theta_3) \end{bmatrix}$$

quedando la matriz del Jacobiano así:

$$\begin{aligned}J[0][0] &= -2 * l\text{Arm} * \cos(\theta_1) * \sin(\theta_2); \\ J[0][1] &= -2 * l\text{Arm} * \sin(\theta_1) * \cos(\theta_2); \\ J[0][2] &= 0;\end{aligned}$$

$$\begin{aligned}J[1][0] &= (2 * l\text{Arm} * \sin(\theta_1 - \theta_3)); \\ J[1][1] &= 0; \\ J[1][2] &= -l\text{Columna1} * \sin(\theta_3) - l\text{Columna2} * \sin(\theta_3) - \\ &\quad 2 * l\text{Arm} * \sin(\theta_1 - \theta_3);\end{aligned}$$

$$\begin{aligned}
 J[2][0] &= 2 * l_{Arm} * \cos(\theta_1 - \theta_3) * \cos(\theta_2); \\
 J[2][1] &= -2 * l_{Arm} * \sin(\theta_1 - \theta_3) * \sin(\theta_2); \\
 J[2][2] &= l_{Columna1} * \cos(\theta_3) + l_{Columna2} * \cos(\theta_3) - \\
 &\quad (2 * l_{Arm} * \cos(\theta_1 - \theta_3)) * \cos(\theta_2);
 \end{aligned}$$

Brazo Derecho.

La expresión analítica del **Brazo Derecho** es igual a la de su brazo izquierdo. Entonces lo único que cambia son los valores en sus ángulos ($\theta_1, \theta_2, \theta_3$)

$$\begin{aligned}
 \theta_1 &= \text{hum} \rightarrow x_{RotBrazo2} / 57.295f; \\
 \theta_2 &= \text{hum} \rightarrow y_{RotBrazo2} / 57.295f; \\
 \theta_3 &= \text{hum} \rightarrow x_{RotCodo2} / 57.295f;
 \end{aligned}$$

Posiciones de los Brazos.

Y las posiciones (X_0, Y_0, Z_0) para los brazos al calcular la ecuación (6.6.1) en sus respectivos ángulos son:

Movimiento del Brazo Derecho (Rojo) hacia atrás,

$$y_{RotBrazo2} = 0.0, z_{RotBrazo2} = 0.0$$

X0	Y0	Z0	XrotBrazo2 θ_1	XRotCodo2 θ_3
-20.0	-22.15	-8.32	-13.93	10.60

Movimiento del Brazo Izquierdo (Azul) hacia delante,

$$y_{RotBrazo2} = 0.0, z_{RotBrazo2} = 0.0$$

X0	Y0	Z0	XrotBrazo2 θ_1	XRotCodo2 θ_3
-20.0	-19.87	6.23	-14.61	89.66

Al realizar un análisis sobre el sistema coordinado de la mano izquierda, el brazo se encuentra que esta en dirección z . Por lo tanto los ángulos θ_1, θ_3 del brazo rotan sobre el eje x (figura 6.5) [SJ99] y θ_2 es considerado igual a cero. Entonces se puede obtener el movimiento de los brazos que pueden estar hacia atrás o hacia delante y generar así una pose inicial. Usando cinemática inversa (ver anexo B y C) y interpolación para obtener sus poses intermedias hasta llegar a una meta final, dando el movimiento de brazos

6.6.2. Expresión Analítica de las Piernas.

A continuación se describe la expresión analítica de la **Pierna Izquierda** para generar su movimiento

Pierna Izquierda.

A partir de la (figura 6 5) [SJ99] y usando trigonometría, se obtienen las expresiones analíticas de la ecuación (6 6 2). Donde los valores para los ángulos ($\theta_1, \theta_2, \theta_3$) de la pierna izquierda son:

$$\theta_1 = \text{ángulo pierna Izquierda en X} = \text{hum} \rightarrow x \text{RotCadera} 1/57.295f;$$

$$\theta_2 = \text{ángulo pierna Izquierda en Y} = \text{hum} \rightarrow y \text{RotCadera} 1/57.295f;$$

$$\theta_3 = \text{ángulo Rodilla Izquierdo en X} = \text{hum} \rightarrow x \text{RotRodilla} 1/57.295f;$$

y los valores para las variables de esta ecuación son:

$$w_{\text{Pelvis}} = 20, l_{\text{Pierna}} = 60.$$

Así las expresiones analíticas obtenidas dan la siguiente ecuación:

$$X_0 = -w_{\text{Pelvis}}/2 - (l_{\text{Pierna}} * \sin(\theta_3 + \theta_1) + l_{\text{Pierna}} * \sin(\theta_1)) * \sin(\theta_2);$$

$$Y_0 = -30 - (l_{\text{Pierna}} * \cos(\theta_3 + \theta_1) + l_{\text{Pierna}} * \cos(\theta_1)); \quad (6.6.2)$$

$$Z_0 = (l_{\text{Pierna}} * \sin(\theta_3 + \theta_1) + l_{\text{Pierna}} * \sin(\theta_1)) * \cos(\theta_2);$$

Y la construcción del Jacobiano de la expresión anterior es:

$$J[0][0] = -(l_{\text{Pierna}} * \cos(\theta_3 + \theta_1) + l_{\text{Pierna}} * \cos(\theta_1)) * \sin(\theta_2);$$

$$J[0][1] = -(l_{\text{Pierna}} * \sin(\theta_3 + \theta_1) + l_{\text{Pierna}} * \sin(\theta_1)) * \cos(\theta_2);$$

$$J[0][2] = -l_{\text{Pierna}} * \cos(\theta_3 + \theta_1) * \sin(\theta_2);$$

$$J[1][0] = -(l_{\text{Pierna}} * \sin(\theta_3 + \theta_1) - l_{\text{Pierna}} * \sin(\theta_1));$$

$$J[1][1] = 0;$$

$$J[1][2] = l_{\text{Pierna}} * \sin(\theta_3 + \theta_1);$$

$$J[2][0] = (IPierna * \cos(\theta_3 + \theta_1) + IPierna * \cos(\theta_1)) * \cos(\theta_2);$$

$$J[2][1] = -(IPierna * \sin(\theta_3 + \theta_1) + Pierna * (\sin(\theta_1)) * \sin(\theta_2));$$

$$J[2][2] = (IPierna * \cos(\theta_3 + \theta_1)) * \cos(\theta_2);$$

Pierna Derecha.

Las expresiones analíticas de la pierna derecha son iguales a los de la pierna izquierda Los ángulos para la ecuación (6.6.2) son:

$$\theta_1 = \text{hum} \rightarrow x_{\text{RotCadera2}} / 57.295f;$$

$$\theta_2 = \text{hum} \rightarrow y_{\text{RotCadera2}} / 57.295f;$$

$$\theta_3 = \text{hum} \rightarrow x_{\text{RotRodilla2}} / 57.295f;$$

Posiciones de las Piernas.

Y las posiciones (X0, Y0, Z0) para las piernas al calcular la ecuación (6.6.2) en sus respectivos ángulos son:

Movimiento de la Pierna Derecha (Rojo) hacia atrás,
 $y_{\text{RotCadera2}} = 0.0, z_{\text{RotCadera2}} = 0.0$

X0	Y0	Z0	XrotCadera2 θ1	XrotRodilla2 θ3
-10.0	-85.08	18.43	33.0	-29.0

Movimiento de la Pierna Izquierda (Azul) hacia delante,
 $y_{\text{RotCadera1}} = 0.0, z_{\text{RotCadera1}} = 0.0$

X0	Y0	Z0	XrotCadera1 θ1	XrotRodilla1 θ3
-10.0	-69.24	-42.82	-33.0	-29.0

Realizando un análisis sobre el sistema coordenado de la mano izquierda, se encuentra que esta en dirección z Por lo tanto los ángulos θ1, θ3 de las piernas rotan sobre el eje x (figura 6.5) [SJ99], y θ2 es considerado igual a cero, con esto se puede obtener el movimiento de las piernas y pueden estar hacia atrás o hacia delante generando así su pose inicial. Usando cinemática inversa (ver anexo B y C) e interpolación, obtiene sus poses intermedias hasta llegar a una meta final, dando el movimiento de las piernas

6.6.3 Cinemática Inversa del Pie.

La fase de apoyo inicia después de que el talón esta rigurosamente en la tierra Después a mayor duración un pie se aleja de la tierra y el otro permanece en la tierra hasta el final de la fase de apoyo, para comenzar la fase de doble apoyo que se da cuando los dos pies se alojan sobre la tierra plana

La postura del pie puede ser determinada por dos parámetros:

1 - El ángulo lugar - sobre en el comienzo de la fase de apoyo, entre el pie de apoyo y tierra, para apoyar el peso es considerado como cero

$$h \rightarrow xRotTobillo1 = 0$$

$$h \rightarrow xRotTobillo2 = 0$$

2 - El ángulo levantamiento o de despegue vertical, al final de la fase de apoyo desde la tierra para el giro de la pierna Entonces la cinemática inversa calcula los ángulos de unión de la cadera y rodilla. Finalmente el ángulo levantamiento (θ) se calcula, sumando sus respectivas uniones de la cadera y rodilla

$$h \rightarrow xRotTobillo1 = XrotCadera1 + XrotRodilla1;$$

$$h \rightarrow xRotTobillo2 = XrotCadera2 + XrotRodilla2;$$

Movimiento del Pie Derecho (Rojo), levantamiento

$h \rightarrow xRotTobillo2$	$XrotCadera2 \theta1$	$XrotRodilla2 \theta3$
4.0	33.0	-29.0

Movimiento dl Pie Izquierdo (Azul), lugar -sobre

$h \rightarrow xRotTobillo1$	$XrotCadera1 \theta1$	$XrotRodilla1 \theta3$
0	-33.0	-29.0

Una vez que los dos ángulos del pie, lugar - sobre y levantamiento, son determinados, la postura del pie esta definida. Y cambia dependiendo la posición del pie, de levantamiento a lugar sobre y viceversa

6.7 Explicación del Movimiento.

Las ecuaciones anteriores son la parte principal para la animación del esqueleto en caminado estándar, dependen de las condiciones de posicionado ($X0$, $Y0$, $Z0$) de cada uno de ellos (el brazo izquierdo, el brazo derecho, la pierna izquierda, la pierna derecha) y en cada caso nos dará un distinto tipo de movimiento hasta llegar a la meta que cada uno persigue, ver figura 6.1 y 6.2

Movimiento de los Hombros.

El movimiento de los hombros es en sentido contrario al movimiento de la pelvis y dependiendo de la situación este puede tener diferente valor: El desplazamiento de hombros lateral, en un caminado estándar involucra el desplazamiento de los hombros de lado a lado

Hum->RotPelvis - = 1.6;

Hum->RotPelvis + = 1.6;

Movimiento de la Cadera y Pelvis.

La rapidez de la pierna de giro durante la fase de giro, siendo más rápida generalmente durante la mitad y más lenta al principio y terminado de la fase de giro[SJ99]

La pelvis se mueve hacia delante con poca variancia en rapidez, siendo más rápida durante la fase de apoyo doble y más lenta en la mitad de la postura de fase. La rapidez de la pelvis varía un poco cuando esta se mueve adelante, durante la fase de apoyo de la pierna y es más rápida durante la fase de apoyo doble.

La rotación pélvica acerca el tronco del cuerpo alternativamente a la izquierda y a la derecha, relativa a la dirección del caminado. Como un índice para animación Saunders et al [SJ99] da ± 3 grados para la amplitud en pasos de caminado estándar.

La inclinación pélvica en pasos de caminado estándar, debido a la inclinación de la pelvis (± 5 grados acerca el eje de la dirección de caminado) la cadera de la pierna de giro cae ligeramente bajo la cadera de la pierna de apoyo. Para introducir la inclinación de la pelvis, esta necesita ser cuidadosamente implementada porque puede poner en peligro la posibilidad de que la pierna de giro penetre en la superficie de la tierra. Para compensar este efecto deseado, una flexión de rodilla de la pierna de giro es necesaria.

Desplazamiento pélvico lateral, el caminado estándar involucra desplazamiento pélvico de lado a lado y incrementa o decrecienta dependiendo del valor de la parte izquierda o derecha.

Hum->RotPelvis += 1.6;

Hum->RotPelvis -= 1.6;

Movimiento total del cuerpo.

La parte importante para controlar el movimiento total del esqueleto es el tronco y para trasladarlo sobre z, se incrementa la variable (avance) en:

Hum->Avance + = 1.09;

6.8 Animación.

El proveer un esquema de animación con respecto a movimientos de caminado humano en ambientes virtuales es una parte del objetivo de este trabajo.

El esqueleto es animado variando los ángulos de rotación locales aplicadas con el tiempo a cada unión y controlando las transformaciones que cambian con el tiempo. Un esqueleto puede animarse variando las rotaciones locales aplicadas con el tiempo a cada ángulo, así como la translación global aplicada al ángulo de la raíz. La especificación del movimiento y el problema del control es eso de dirigir la manera en que estas transformaciones cambian con el tiempo.

El describir los movimientos básicos de la figura articulada en su caminado requirió de un proceso iterativo que comienza con un paso continuando sobre una serie de pasos usando cinemática para el cambio de ángulos y posiciones del resultado final en brazos, piernas y pies, para generar las poses necesarias que construyen la animación. Para la ejecución del programa se realiza lo siguiente:

Escribe Ky Nombre_archivo

Para ver la animación se escribe `igl -sync 30 Nombre de archivo`

Otra forma es crear las poses de un paso, estas se generan así

Escribe Ky `rendrib`

Empieza a generar archivos cuyo nombre del archivo es así

`Kn#numero.tif`

Estos archivos `tif` se apoyan de un programa llamado `paint shop pro 7.0` que une las poses y simula la ilusión del movimiento del caminar humano representado por la figura articulada.

En el anexo A se encuentra el diagrama de bloques que explica el proceso que se llevo a cabo para lograr la animación, en el anexo B un diagrama de flujo del algoritmo de animación y una explicación breve de sus módulos y finalmente su código del programa en el anexo C

El programa se desarrolló en Visual C++ para ambiente Windows98 bajo la plataforma de interfaz RenderMan (lenguaje capaz de modelar el comportamiento de las superficies, fuentes de luz) empleando un programa de rendering para generar imágenes de alta calidad llamado BMRT2.6 (Blue Moon Rendering Tools versión 2.6) que contiene la biblioteca libribout lib

Caminado de la Figura Articulada.

El resultado de este trabajo se muestra en algunos cuadros que pueden ser comparativos con los caminados humanos descritos en el capítulo 3

La animación la define una secuencia de movimientos, así una animación compleja la define una secuencia de movimientos combinados

La animación se explica a través del ciclo del caminado, donde cada vez que das un paso, avanzas hacia delante inclinando ligeramente el cuerpo y como la pierna diestra atrapa el cuerpo para prevenir a este de caerse Después de que el pie toca la tierra, el peso del cuerpo es transferido al pie y la rodilla doblada absorbe el choque La pierna de frente entonces levanta el cuerpo y impulsa el cuerpo hacia delante y como la pierna de atrás gira hacia arriba para atrapar el cuerpo otra vez y el ciclo se repite

Esta descripción de los distintos movimientos del modelo (figura articulada), basada en cinemática, para alcanzar sus metas y sus restricciones ayudan a producir una animación compleja creíble.

Esto muestra lo obtenido por el algoritmo de animación usando cinemática, aunque parece simple no lo es, se requiere de bastante estudio y paciencia para lograr un resultado convincente de una figura articulada humana

Se tomaron solo algunas poses del caminado para demostrar los resultados obtenidos de este trabajo, los cuales se pueden comprobar con los caminados grabados anteriormente del capítulo 3 y se muestran a continuación:

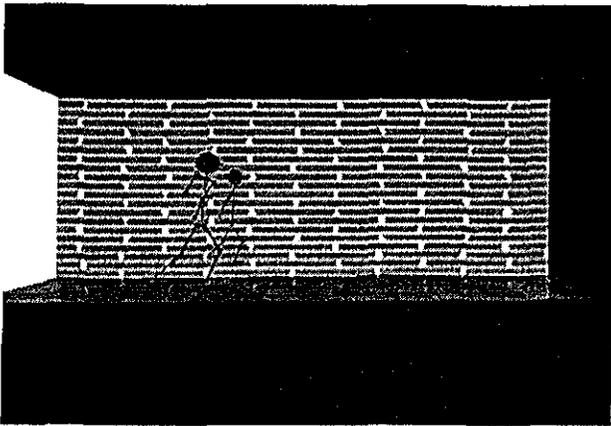


Figura 6.6 Pose: Pierna izquierda apoyo, kn#131.tif

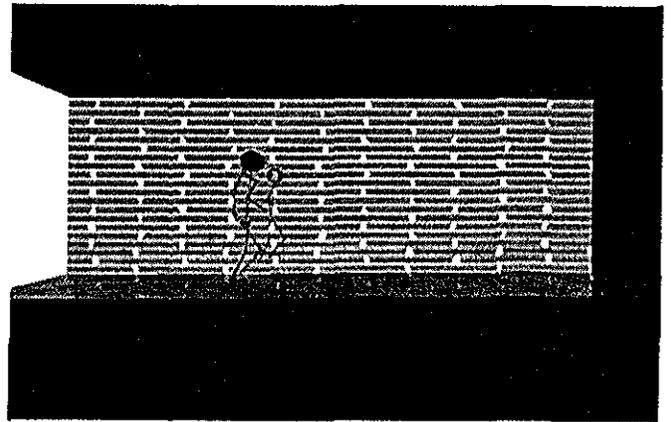


Figura 6.7 Pose: Pierna de giro derecha, kn#145.tif

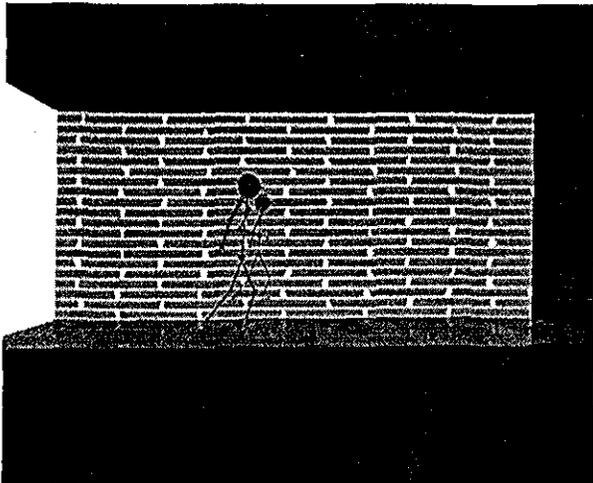


Figura 6.8 Pose: fase de doble apoyo kn#159.tif

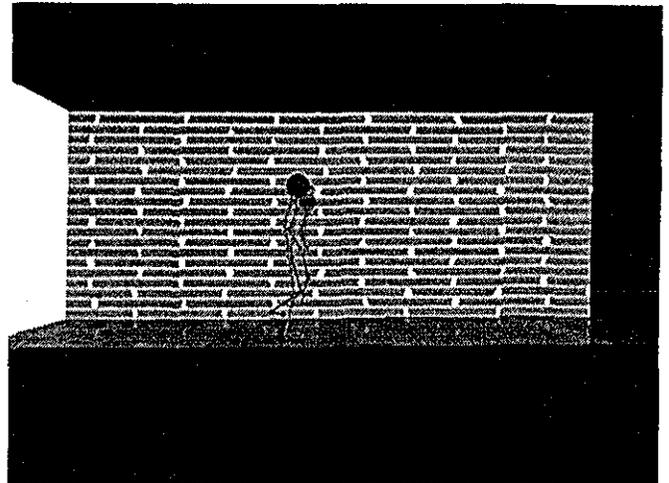


Figura 6.9 Pose: fase de mitad postura kn#169.tif

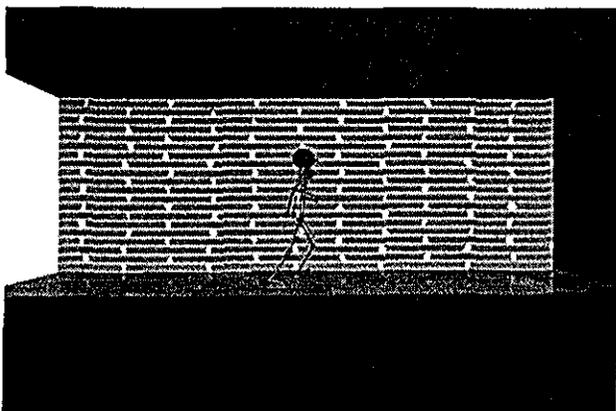


Figura 6.10 Pose: fase de apoyo doble kn#185.tif

TESIS CON
FALLA DE ORIGEN

Capítulo 7.

Un Análisis mas avanzado: Generación automática de pasos.

Una de las de acciones más rutinarios que realizan los humanos es caminar. También una de las acciones más importantes es su animación. A la fecha, sin embargo, una herramienta automatizada para generar los pasos humanos no está disponible. La dificultad se levanta en dos niveles: en el nivel bajo, la mayoría de los métodos no pueden generar muchas variaciones necesarias de pasos para cumplir las obligaciones funcionales de caminar: caminando a lo largo de una trayectoria arbitraria en el terreno posiblemente desigual. En el nivel alto, le exigen al animador que controle el caminando manipulando los parámetros de bajo nivel; esto es una tarea difícil y tediosa. Este capítulo trata el análisis de los elementos que toman en cuenta para crear una herramienta que genera los pasos humanos. El material descrito se basa en la referencia [SM2001].

7.1 Antecedentes.

Para que una herramienta sea útil, un sistema de animación de pasos debe ser bastante poderoso para crear la curva de locomoción (el caminado a lo largo de una trayectoria curva) en el terreno desigual. Además, el animador no necesita poner parámetros de bajo nivel al sistema de pasos; el sistema de caminar debe calcularlos, para que el animador sea libre de generar una tarea más creativa y estimulante.

La meta es construir una herramienta que releve la carga de animar el caminado ocupándose de los detalles de nivel bajos para la generación del movimiento así como de los detalles de niveles más altos para poner los parámetros correctamente.

Este capítulo enfoca el problema de la generación de pasos a través de tres módulos, ElevWalker, basado en la representación del ángulo de elevación sagital, calcula los pasos de nivel bajo y puede generar la curva de locomoción. ElevInterp, presenta una nueva solución al problema de interpolación de movimiento inverso, realiza la interpolación del conjunto de datos para controlar la longitud y la altura del paso. MetaGait usa datos biomecánicos empotrados y reglas para controlar los parámetros de los componentes anteriores. El sistema puede usarse para realizar ambos: la animación tradicional de pasos humana fuera de la línea así como la animación en línea de una animación autónoma o por el usuario que controla a un agente en una aplicación interactiva.

7.2 Modelo de Cinemática.

Todavía ha habido buen éxito usando cinemáticas. Esto principalmente es debido al talento de animadores experimentados, así como la creciente facilidad de captura cinemática de datos y usando éstos para manejar la animación.

El problema al usar las cinemáticas para los movimientos generadores del caminado ha sido al generalizar éstos datos de la cinemática a diferentes figuras y situaciones diferentes: estilo de caminado, trayectoria y terreno. Bruderlin y Calvert con las cinemáticas inversas aseguran que el balance del tobillo siga una trayectoria especificada.



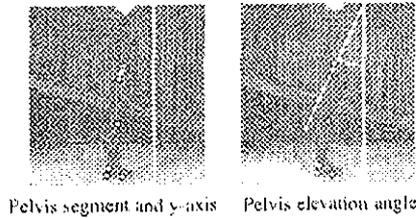
Figura 7.1 Estructura del modelo para aplicar cinemática.

El generador de pasos calcula el movimiento del cuerpo más bajo (cadera, rodilla, tobillo, pie (figura 7.1) [SM2001] de una figura humana, por lo tanto la cadena de cinemática sólo incluye la pelvis, las piernas y los pies.

La cinemática también calcula la posición y orientación de la raíz.

7.3 ElevWalker: La generación del movimiento de pasos.

ElevWalker presenta, un nuevo generador de pasos de nivel bajo basado en los ángulos de elevación sagital, que permite curvas de locomoción el caminado a lo largo de una trayectoria curva ElevWalker introduce un nuevo algoritmo usando la representación del movimiento del ángulo de elevación sagital que permite generar la curva de locomoción.



TESIS CON
FALLA DE ORIGEN

Figura 7.2 Primer cuadro el eje Y y el segmento de la pelvis, el segundo el ángulo de elevación de la pelvis. Midiendo el ángulo de elevación sagital de la pelvis.

ElevWalker toma los datos del movimiento en forma de ángulos de elevación sagital El ángulo θ , es su ángulo de elevación sagital, (figura 7 2) [SM2001] entonces:

$$\tan \theta = (v_x^{(sag)}) / -v_y^{(sag)} \quad (1)$$

ElevWalker calcula los ángulos de unión del cuerpo más bajo: el pie, la pierna más baja, la pierna superior y la pelvis (figura 7 3) [SM2001]. También calcula la cinemática (o figura) de la raíz.

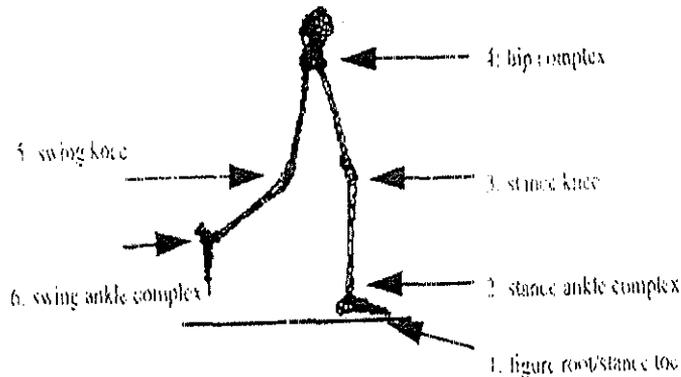


Figura 7.3 Los ángulos de unión y su orden de cálculo.

La orientación del plano sagital, se parametriza solo por el ángulo ψ (figura 7.4) [SM2001] y usa ψ después para realizar la curva de locomoción.



Figura 7.4 El primer cuadro coordenadas globales, el segundo coordenadas sagitales. El plano sagital.

ElevWalker tiene un algoritmo para calcular la cinemática, la transformación de la raíz y los ángulos de unión que comienzan en la posición actual del pie. La transformación de la raíz de la figura se calcula primero. Luego calculamos los ángulos de unión de la figura, preparando la posición de la pierna, a través de la pelvis y echar atrás la pierna de giro; el orden de estos cálculos es (figura 7.3) [SM2001]:

1. Raíz de la figura / posición del pie.

La cinemática primero calcula la raíz de la figura en cada paso de tiempo. En todo momento, la raíz de la figura se localiza dentro de la posición actual del pie (figura 7.5) [SM2001]

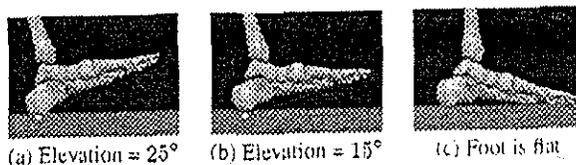


Figura 7.5 El pie rota cerca del punto de contacto (en blanco) hasta que este es plano sobre la tierra.

Calcula el punto instantáneo de contacto del pie con la tierra. Una vez que tiene el punto de contacto, gira el segmento del pie y recalcula el punto de contacto en cada frame, el pie se queda en la superficie de la tierra y no la penetra. El pie plano esta sobre la tierra cuando un punto en el pie delantero (segmento del dedo del pie) y un punto en el pie posterior (el segmento del talón) están los dos en contacto con la tierra. Cuando esta condición se encuentra, la raíz se mueve a los dedos de los pies y la elevación del pie es entonces satisfecha por la primera unión del dedo del pie (permitiendo al talón subir desde la tierra). La transformación de la raíz ya no cambia hasta que el lado de giro y el lado de la posición se intercambien.

2. Posición compleja del tobillo.

Definen la anchura de la posición $Wst(t)$ a ser la distancia de la posición del pie a la pelvis, en la dirección del plano sagital normal. La posición compleja del tobillo, resuelve dos restricciones simultáneamente: la restricción del ángulo de elevación de la pierna más baja y lo siguiente:

$$(ast - hst) \cdot ctrl \cdot n - Wst(t) \quad (8)$$

donde $ctrl \cdot n$ es un vector normal unitario al plano sagital, ast es un sitio sobre la posición del tobillo y hst es un sitio en la posición de la cadera. Manipulando $Wst(t)$, puede cambiar la anchura lateral de la posición durante los pasos.

Estas restricciones se resuelven para producir los ángulos de unión superiores y más bajos del tobillo simultáneamente. Los ángulos de unión se calculan usando un método de búsqueda numérico: han encontrado que el método de Gauss-Newton es sumamente eficaz, usando los ángulos de unión anteriores como la búsqueda del punto de partida.

3. Posición de la rodilla.

Calcular los ángulos de unión, preparando la posición de la pierna y luego la pierna de giro hacia abajo. En muchas de estas uniones, nos preocupamos por sólo un DOF y una sola restricción del ángulo de elevación. Por ejemplo, la unión de la posición de la rodilla tiene solamente un DOF. Se desea calcular el ángulo de unión θ tal que, el ángulo de elevación sagital del miembro del segmento v produce el conjunto de datos deseados del ángulo de elevación ϕ (figura 7.6) [SM2001]

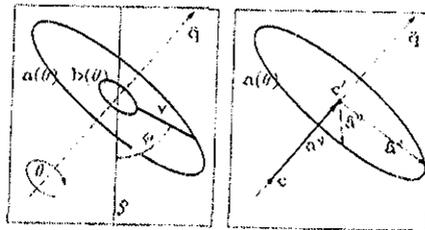


Figura 7.6 Resolviendo la restricción del ángulo de elevación para una unión única.

4. Cadera compleja.

Lo complejo de la cadera consiste en las uniones de la posición de la cadera y la cadera de giro. Tres restricciones de ángulo de elevación aplican a las uniones de la cadera (figura 7.7) [SM2001]: a) la posición de la pelvis, b) el giro de la pelvis y c) los ángulos de elevación superiores de la pierna para el giro. Se agregan restricciones que tiene parámetros de alto nivel.

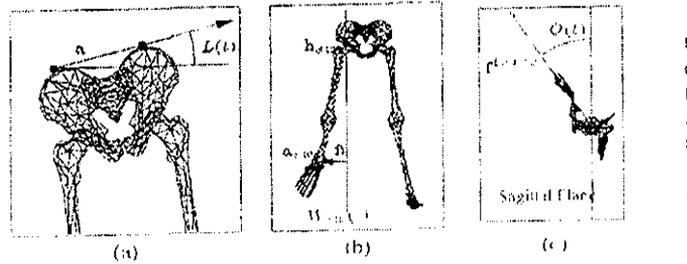


Figura 7.7 El modelo de la cadera controla la lista pélvica, la anchura de giro y el dedo del pie afuera.

La primer restricción parametrizada es para la lista pélvica. Ésta es la cantidad de rotación en el plano frontal y manifiesta como una cadera es más alta que la otra. La lista pélvica es un ángulo cuyo parámetro es $L(t)$ y su restricción asociada es la siguiente:

$$\tan^{-1}(ax, ay) - L(t) = 0$$

Manipulando $L(t)$, se puede cambiar la cantidad de meneo de la cadera durante el paso.

La segunda restricción parametrizada es la anchura de giro. El parámetro para la anchura de giro es $Wsw(t)$ y la restricción es la siguiente:

$$(asw - hsw) * \text{ctrl. n} - Wsw(t) = 0$$

donde asw es un sitio en el tobillo de giro y hsw es un sitio en la cadera de giro.

La tercer restricción parametrizada es el dedo del pie afuera. El dedo del pie fuera es la cantidad de rotación interior / exterior de la pierna: sólo se define el dedo del pie afuera para la pierna de giro, cuando la posición de la pierna es fija en la tierra. El dedo del pie afuera es un ángulo cuyo parámetro es $O(t)$ y su restricción asociada es la siguiente:

$$\cos^{-1}(f^{(trans)} * x^{(sag)}) - O(t) = 0$$

Manipulando $O(t)$, se puede cambiar la dirección de los puntos del pie de giro, en el plano transversal. El dedo del pie afuera se usa para producir la curva de locomoción.

Como en el modelo de la posición del tobillo, se usa un método de búsqueda numérico para encontrar el ángulo de unión de la cadera. Se usa otra vez la solución de los pasos del tiempo anterior como el punto de partida y se encuentra normalmente que el método converge rápidamente, dentro de 4 iteraciones Gauss-Newton.

5. Rodilla de giro.

Calcular los ángulos de unión, preparando la posición de la pierna y luego la pierna de giro hacia abajo. Calcular el ángulo de unión analíticamente - la posición del dedo del pie, la rodilla de giro y el tobillo de giro. Se desea calcular el ángulo de unión θ tal que, el ángulo de elevación sagital del miembro del segmento v produce el conjunto de datos deseados del ángulo de elevación ϕ .

6. Tobillo complejo de giro.

En la posición compleja del tobillo, tenemos dos DOFs, las uniones superiores y más bajas del tobillo y sólo una restricción de elevación en la posición más baja de la pierna.

Calcular los ángulos de unión, preparando la posición de la pierna y luego la pierna de giro hacia abajo. Calcular el ángulo de unión analíticamente - la posición del dedo del pie, la rodilla de giro y el tobillo de giro -

Se desea calcular el ángulo de unión θ tal que, el ángulo de elevación sagital del miembro del segmento v produce el conjunto de datos deseados del ángulo de elevación ϕ .

Como en el modelo de la posición del tobillo, usamos un método de búsqueda numérico para encontrar el ángulo de unión de la cadera. Se usa otra vez la solución de los pasos del tiempo anterior como el punto de partida y se encuentra normalmente que el método converge rápidamente, dentro de 4 iteraciones Gauss-Newton.

Curva de locomoción.

El encabezamiento de la dirección ψ describe la orientación del plano sagital y la dirección del caminado (figura 7.4) [SM2001]. Dejando esto fijo, la figura camina en una línea recta. Sin embargo, si variamos ψ resultan curvas de locomoción. Esto es sumamente importante para reducir el número de conjunto de datos necesarios. Tan sólo modificando el parámetro ψ genera la curva de locomoción y no requiere el coleccionar más datos del movimiento.

7.4 ElevInterp: Transformación de datos del movimiento.

ElevInterp, usa un nuevo algoritmo de interpolación de movimiento inverso para ocuparse de la locomoción del terreno desigual. Utiliza una nueva técnica para resolver el problema de interpolación del movimiento inverso calculando los parámetros de la interpolación de bajo nivel puesto que los usuarios sólo trabajan con los parámetros de alto nivel la longitud y la altura del paso. Usa el conjunto de datos múltiples e interpola entre ellos para generar los nuevos conjuntos de datos que satisfacen los requisitos de longitud y de altura del paso. Éstos se dan entonces a ElevWalker. ElevInterp entonces controla la longitud del paso.

Sin embargo, la interfaz ElevInterp permite a los controladores de alto nivel usar como parámetros de entrada la longitud y altura del paso. Calcula qué conjunto de datos interpolar y cómo mezclarlos, lograr la longitud y altura especificada, eliminando la necesidad del controlador (o animador) para especificar estos a bajo nivel y frecuentemente los parámetros no intuitivos. Para esta tarea, han desarrollado una nueva, solución eficiente a este problema de interpolación de movimiento inverso.

7.4.1 Interpolación del movimiento inverso.

Sé penso primero sobre los movimientos como conjuntos de datos. Para caracterizar un movimiento, se miden algunas características cuando este se aplica a una figura específica, como la longitud y altura del paso. Permite a $f(ds)$, que sea una función que mide la altura y la longitud del paso para lograr un movimiento (un conjunto de datos). Esto produce un

espacio de 2 dimensiones, que en cada conjunto de datos, puede localizarse un espacio de altura / de longitud

También se piensa en conjuntos de datos del movimiento como los puntos abstractos en un espacio de conjuntos de datos Para ser consistente con las dos características que esta midiendo, el espacio del conjunto de datos tiene también dos dimensiones. Las coordenadas en este espacio servirán después como los parámetros de interpolación, así los hemos situado a ellos a tener aproximadamente la longitud del paso en el rango [0,1] y la altura del paso en el rango [-1, 1]

Estos dos espacios (figura 7 8) [SM2001] El espacio del conjunto de datos y el espacio de altura / de longitud estas coordenadas representan las distancias reales en centímetros. $f(ds)$ toma un conjunto de datos desde un espacio del conjunto de datos (la interpolación) al espacio fisico.

Cuando un animador, o MetaGait, demanda que ElevInterp genere un paso con la longitud SL y altura SH, ElevInterp debe determinar qué conjunto de datos interpolar y que pesos deben ser de la interpolación

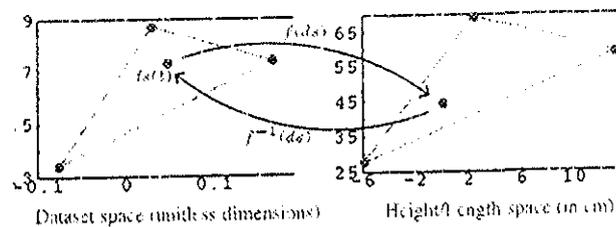


Figura 7.8 El primer cuadro, espacio del conjunto de datos (sin unidad de dimensión), el segundo espacio (en cm) altura/longitud. Mapeando desde el espacio del conjunto de datos al espacio fisico.

Éste es el problema de interpolación de movimiento inverso, porque debe invertir ambos la función de medida $f(ds)$ así como la función de la interpolación

ElevInterp realiza una interpolación que usa todavía una simple armazón general: las coordenadas de triangulaciones y baricéntricas En el sistema, usan los vértices que contiene el triángulo Permite a las coordenadas baricéntricas de este conjunto de datos con respecto a los vértices en espacio altura / longitud ser (a, b, c). Una interpolación baricéntrica puede definirse como la siguiente:

$$ds(t) = a * ds1(t) + b*ds2(t) + c*ds3(t) \quad (9)$$

donde $ds(t)$ es el nuevo conjunto de datos y $ds1(t)$, $ds2(t)$ y $ds3(t)$ son los conjuntos de datos asociados con los vértices

Las coordenadas baricéntricas dadas, son simples para calcular un nuevo conjunto de datos por la vía de interpolación baricéntrica

Invirtiendo $f(ds)$, la función que mapea los conjuntos de datos físicos a reales (la longitud / la altura del paso), toma un poco más de esfuerzo. Primero nosotros daremos nuestra

definición de $f(ds)$. Para estimar la longitud y altura del paso al final del ciclo del paso, se usa un modelo planar de la figura. Permita que la posición de los ángulos de elevación laterales sea, $ds(0.5) = \{\phi^{st}_1, \phi^{st}_2, \phi^{st}_3, \phi^{st}_4\}$, y el giro de los ángulos de elevación laterales son $ds(1.0) = \{\phi^{st}_1, \phi^{st}_2, \phi^{st}_3, \phi^{st}_4\}$

También requiere las longitudes de los miembros. Permite a L tener las longitudes de los miembros desde los pies a la pelvis: $L = (l_1, l_2, l_3, l_4)$, respectivamente

Entonces la posición estimada del tobillo de giro $\sim psw$ es el siguiente:

$$F(ds; L) = \sim psw \\ = \sum_{i=1}^4 l_i R(\phi^{st}_i)(0,1,0) + \sum_{i=2}^4 l_i R(\phi^{sw}_i)(0,-1,0) \quad (10)$$

donde $R(\phi)$ es una matriz de la rotación 2 por 2.

Este cálculo produce $\sim psw = (l, h)$. La primera suma representa **segmentos de la posición de la pierna** y la segunda suma representa **segmentos del giro de la pierna**. **Invertiendo $f(ds)$ significa encontrar el conjunto de datos que produce la longitud y altura del paso correcto**. Desde el punto (SL, SH) deseado existen en el espacio (longitud / altura), podemos determinar sus coordenadas baricéntricas con respecto a sus contenidos en el triángulo. Se usan estas coordenadas baricéntricas en el espacio del conjunto de datos para calcular los nuevos conjuntos de datos del $ds(t)$.

El algoritmo completo para encontrar los parámetros de interpolación para una combinación (altura / longitud) (SL, SH) deseada es como sigue:

1. (espacio (altura / longitud)) localice el triángulo T que contiene (SL, SH) en espacio (altura / longitud);
2. espacio (altura / longitud) calcula las coordenadas baricéntricas de (SL, SH) con respecto a T ;
3. (Espacio del conjunto de datos) calcula un nuevo conjunto de datos $ds(t)$ de acuerdo a la ecuación 9 que usa las coordenadas baricéntricas del paso 2;
4. Si $f(ds) = (SL, SH)$, entonces el conjunto de datos satisface los requisitos (altura / longitud), vaya a 6
5. Por otra parte, use $ds(t)$ y $f(ds)$ como el punto de partida en la búsqueda numérica;
6. (Espacio en el conjunto de datos y (altura / longitud)) la inserción del conjunto de datos en ambas triangulaciones usa (SL, SH) para las coordenadas físicas y $(SL / longitud de _ la pierna, SH / longitud de _ la pierna)$ para las coordenadas del conjunto de datos.

Los módulos ElevWalker y ElevInterp tienen un pequeño conjunto de parámetros intuitivos de alto nivel, - tales como la longitud del paso, la altura del paso, dirección de la cabeza y el dedo del pie-afuera

7.5 MetaGait.

MetaGait, módulo para controlar un alto nivel, permite a un animador controlar una figura que está caminando, especificando una trayectoria. En resumen éstos módulos son de fácil

uso, en tiempo real. La herramienta de animación totalmente automatizada es conveniente para la animación fuera de línea, para ambientes virtuales y simulación

Han construido un módulo de control a nivel-usuario, MetaGait que automatiza las decisiones de cómo seguir trayectorias y terrenos, cómo calcular el próximo paso y cómo calcular los valores de los parámetros del generador del movimiento. La interfaz le permite al usuario dirigir una figura humana para "seguir esta trayectoria a ese punto" y el sistema tiene el cuidado de los detalles. El sistema proporciona una "caja negra" que hace la cosa correcta, con una interacción mínima necesaria.

MetaGait controla el patrón de caminado. El controlador MetaGait, ha empotrado reglas específicas del caminado y datos de pasos sobre cómo estas variables son interrelacionadas, qué le permiten a este calcular sus valores automáticamente.

MetaGait es el primer módulo de control de alto nivel para la animación del paso humano que combina el control de muchas variables del paso para asegurar que las trayectorias de las curvas sigan en un terreno desigual.

MetaGait un controlador de alto nivel que manipula los parámetros de estos módulos para que un animador pueda dirigir una figura simplemente especificando su trayectoria. En cada frame, MetaGait realiza el calculo siguiente:

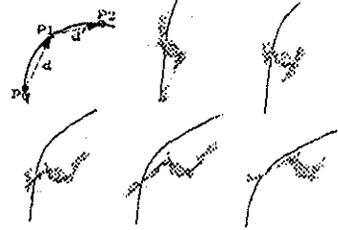
1. Calcula los valores del parámetro para la curva de locomoción.
2. Calcula los valores del parámetro para el terreno desigual, basado en los conjuntos de valores para la curva de locomoción.
3. Notifica al generador del movimiento para usar los nuevos valores del parámetro

7.5.1 Curva de locomoción.

Siguiendo una trayectoria curva especificada por el animador involucra cambios en el parámetro ψ encabezando la dirección, así como modificando posiblemente el dedo del pie afuera y los parámetros de longitud de paso $O(t)$ y $SL(t)$ en cada frame. Además, si la trayectoria contiene un giro muy repentino, podemos requerir regresar un giro para aumentar el que se vuelva al paso normal que usa ElevWalker

Deje que $p(t)$ sea una trayectoria especificada por el usuario. La primera tarea es calcular la dirección de encabezamiento ψ . Una posibilidad simplemente es usar el gradiente $\nabla p(t)$. Sin embargo, este valor puede cambiar muy rápidamente, particularmente en las esquinas repentinamente. Se usa una diferencia del vector que varía menos desordenadamente

En cada frame, estimamos dos puntos: p_0 , el punto en la trayectoria más cerca dónde la figura esta y p_1 , el punto en la trayectoria dónde la figura estará ahora en un paso. Dado estos dos puntos, se forma el vector $d = p_1 - p_0$; d es la dirección que desea encabezar. La (figura 7.9) [SM2001] muestra un ejemplo de estas cantidades. Se calcula la orientación del plano sagital entonces como $\psi = \text{atan2}(-dx, dx)$



TESIS CON
FALLA DE ORIGEN

Figura 7.9 Los puntos estimados usados siguiendo una trayectoria y algunos frames de ejemplo.

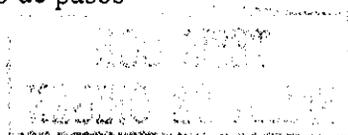
También puede modificar el parámetro el dedo del pie afuera $O(t)$ durante la trayectoria de la curva. Entonces el dedo del pie afuera afecta sólo la pierna de giro, mientras manipulándolo no cambia los pasos durante el paso actual. En cambio, el dedo del pie afuera se modifica en la preparación para el próximo paso.

Una figura de orientación ψ es dependiente en la orientación del pie de la posición actual, porque la cantidad de rotación posible a las caderas está limitado. Entonces la orientación de la posición del pie depende de su orientación durante la fase de giro anterior, se puede manipular su ángulo del dedo del pie afuera durante la fase de giro para hacer un paso subsecuente más fácil.

El algoritmo para esto es como sigue: en cada frame, estimamos la orientación de nuestro cuerpo, d mas arriba. También estimamos la orientación de nuestro cuerpo durante el paso después de este, d' (figura 7.9) [SM2001]. Si la diferencia angular Δ entre estas dos orientaciones es mayor que un umbral T (45° , por defecto). Esto significa que debe ajustar el parámetro del dedo del pie afuera $O(t)$ durante este paso para que el próximo paso fuera más fácil de realizar. Se agrega la diferencia $\Delta - T$ a $O(t)$, apuntando el pie de giro en la dirección de la curvatura.

En la presencia de giros muy repentinos, ajustes grandes del dedo del pie afuera puede parecer antinatural. En estas situaciones, también rota el carácter en la posición del pie. Permita a $T2$ ser un umbral en cuánto podamos ajustar el dedo del pie fuera (25° , por defecto). Si $\Delta - T > T2$, entonces partimos el dedo del pie a $O(t) + T2$. Cuando el próximo paso empieza, se usa un giro para ayudar a que regrese el paso, mientras rotando $(\Delta - T - T2)^\circ$ por un giro en la posición del pie.

Finalmente, la longitud del paso SL también puede cambiar como resultado de la curva de locomoción: uno no toma que la misma longitud del paso al navegar los giros repentinos como al pasear en un camino recto. Se define ψ_{thresh} para ser el ángulo máximo que ψ puede cambiar en un solo ciclo de pasos.



Se calcula una longitud del paso máxima para que el cambio ψ esté a menos de ψ_{thresh} . Para calcular la longitud del paso máximo SL_{max} , iterativamente estimaremos el cambio ψ usando la longitud del paso actual y reduciendo la longitud del paso (actualmente por una cantidad fija) hasta que el cambio sea menor que ψ_{thresh} . Este valor de SL_{max} es usado después, para calcular la longitud del paso real

7.5.2 Locomoción en terreno desigual.

Después que los parámetros para la curva de locomoción han sido calculados. Se calcula la altura y longitud del paso para la locomoción en el terreno desigual. MetaGait calcula la longitud y altura del paso en las coordenadas físicas y permite a ElevInterp ocuparse del problema de calcular los coeficientes de la interpolación para satisfacer estos parámetros. El modelo en el terreno desigual es un proceso iterativo para calcular SH y SL, la altura y longitud del paso actual (figura 7.10) [SM2001]. El proceso consiste en tres pasos:

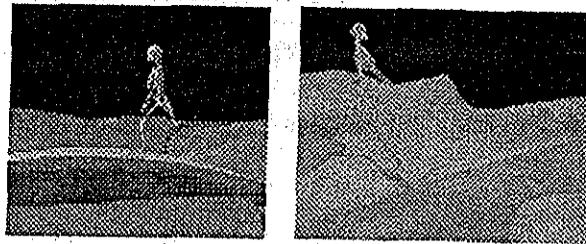


Figura 7.10 Ejemplos de caminados sobre terrenos de pendientes y no planos.

1. Predecir donde aterrizará el pie de giro
2. Encontrar la altura de la tierra en ese punto.
3. Asegurar la longitud del paso en esa altura sea aceptable, por otra parte calcula la nueva longitud y regresa al paso 1.

Para el paso 1, se usa el valor actual de SH y SL como las estimaciones iniciales. Sin embargo, SH y SL miden altura y la longitud del paso en las coordenadas del plano sagital y lo que se necesita es la posición global dónde el pie de giro aterrizará (para el paso 2). La ecuación siguiente transforma éstos en las coordenadas globales:

$$psw = R(\psi)[[SL, SH] + \text{offset}_{\text{pelvis}}] + p_{st}$$

El término $\text{offset}_{\text{pelvis}}$ es la distancia lateral entre las caderas y ayuda a calcular la posición lateral del pie de giro. Para transformar en coordenadas globales se agrega el vector [SL, SH] y se multiplica el vector resultante por $R(\psi)$. Finalmente, se agrega p_{st} , la posición actual del pie, para conseguir la posición global.

En el paso 2, se encuentra la altura de la tierra intersectando una línea vertical que atraviesa psw con la geometría de la tierra. Ésta es sustraída de la posición actual del pie y para producir la diferencia de altura h .

TESIS CON
FALLA DE ORIGEN

Capítulo 7. Un Análisis mas avanzado: Generación automática de pasos.

En el paso 3, se debe probar la altura y la longitud del paso por la compatibilidad. La altura SH y longitud SL del paso son determinadas con respecto al ángulo pendiente de la tierra, p, medido en grados. Las ecuaciones de las pendientes (en los metros) son las siguientes:

$$SL(p) = 0.627 + 0.0075p \text{ (pendiente)} \quad (11)$$

$$= 0.623 - 0.0019p \text{ (ascendente)}$$

$$SH(p) = SL(p) * \tan(p) \quad (12)$$

Debido a la existencia preferida de la longitud del paso para la altura, no se puede escoger simplemente una longitud del paso y calcular la altura correspondiente. La longitud del paso podría necesitar ser alargada o acortarse para ajustar la longitud del paso preferida para esa altura. También se debe asegurar que la longitud del paso no es mayor que la longitud del paso máxima SLmax que calculó para la curva de locomoción.

MetaGait usa las Ecuaciones 11 y 12 para calcular la longitud del paso preferida SLpref para la altura k. SL es igual al min(SLpref, SLmax) y regresa al paso 1. Al final del paso 3, el valor de SH y SL para el próximo paso ha sido determinado.

7.5.3 Determinando los nuevos valores del parámetro.

Después de que los parámetros han sido calculados para la curva de locomoción y el caminado en terreno desigual, se notifica al generador del movimiento para usar los nuevos valores del parámetro. ElevInterp se ocupa de longitud y altura del paso, mientras ElevWalker se ocupa de la dirección del encabezamiento y del dedo del pie afuera.

MetaGait se activa en cada frame, para que estos parámetros estén cambiando constantemente. El generador del movimiento subyacente permite modificar los parámetros en cada frame, mientras permitiéndole ser más sensible a los sistemas que sólo pueden modificar al principio los parámetros de un nuevo paso.

7.6 Resultados.

El sistema se diseñó para la generación del movimiento automatizado en tiempo real, para las aplicaciones de ambiente virtuales y simulaciones. El sistema ha logrado esta meta: para una variedad de ambientes, puede mantener una proporción del frame de 50 + los frames por segundo (ninguna salida gráfica). Estas pruebas fueron hechas en un Pentium 3 de 800 Mhz con 128 Mb de memoria.

Han generado varios ejemplos de caminado en el terreno desigual, usando cinco conjuntos de datos: caminando normalmente, caminado ascendente, caminado en declive, caminado con zancadas cortas (el terreno nivelado), caminando con zancadas largas (el terreno nivelado). Éstos muestran la habilidad de nuestro sistema de ocuparse de la curva de locomoción en una superficie continuamente cambiante que usa una cantidad muy pequeña de datos.

Han usado el sistema para generar la animación en varias figuras diferentes, incluyendo la figura de esqueleto primaria, una figura del esqueleto más pequeña, una figura de esqueleto hembra y una figura humana más grande. Debido al ángulo de elevación sagital la

invariación del conjunto de datos con respecto al tamaño, los mismos conjunto de datos se usaron para todas las tres figuras sin modificación o preprocesamiento.

También compararon los movimientos generados contra aquellos movimientos de los humanos reales. Abajo, comparan las trayectorias directas de la cadera actual, rodilla y tobillo con las trayectorias generadas (figura 7.11) [SM2001] Las trayectorias son muy similares

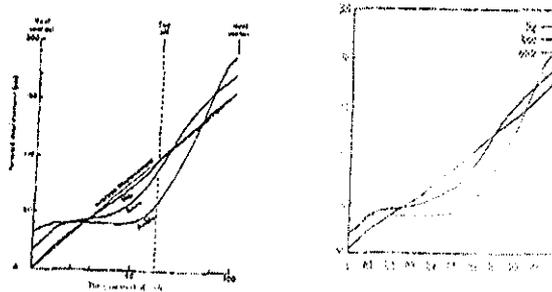


Figura 7.11 Datos de las posiciones de la cadera, rodilla y tobillo vs Datos generados.

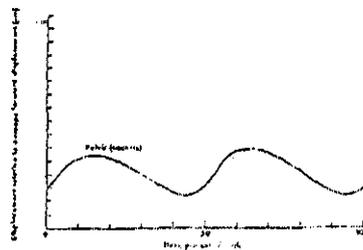


Figura 7.12 Datos de la posición de la pelvis relativa a la velocidad promedio.

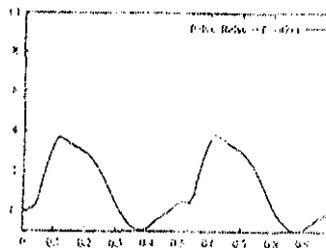


Figura 7.13 Genera la posición de la pelvis relativa a la velocidad promedio.

También comparan la posición directa relativa de la pelvis (figuras 7.12 y 7.13) [SM2001]. Aunque la magnitud es más grande, las formas globales de las trayectorias son muy similares

7.7 Discusión y Trabajo Futuro.

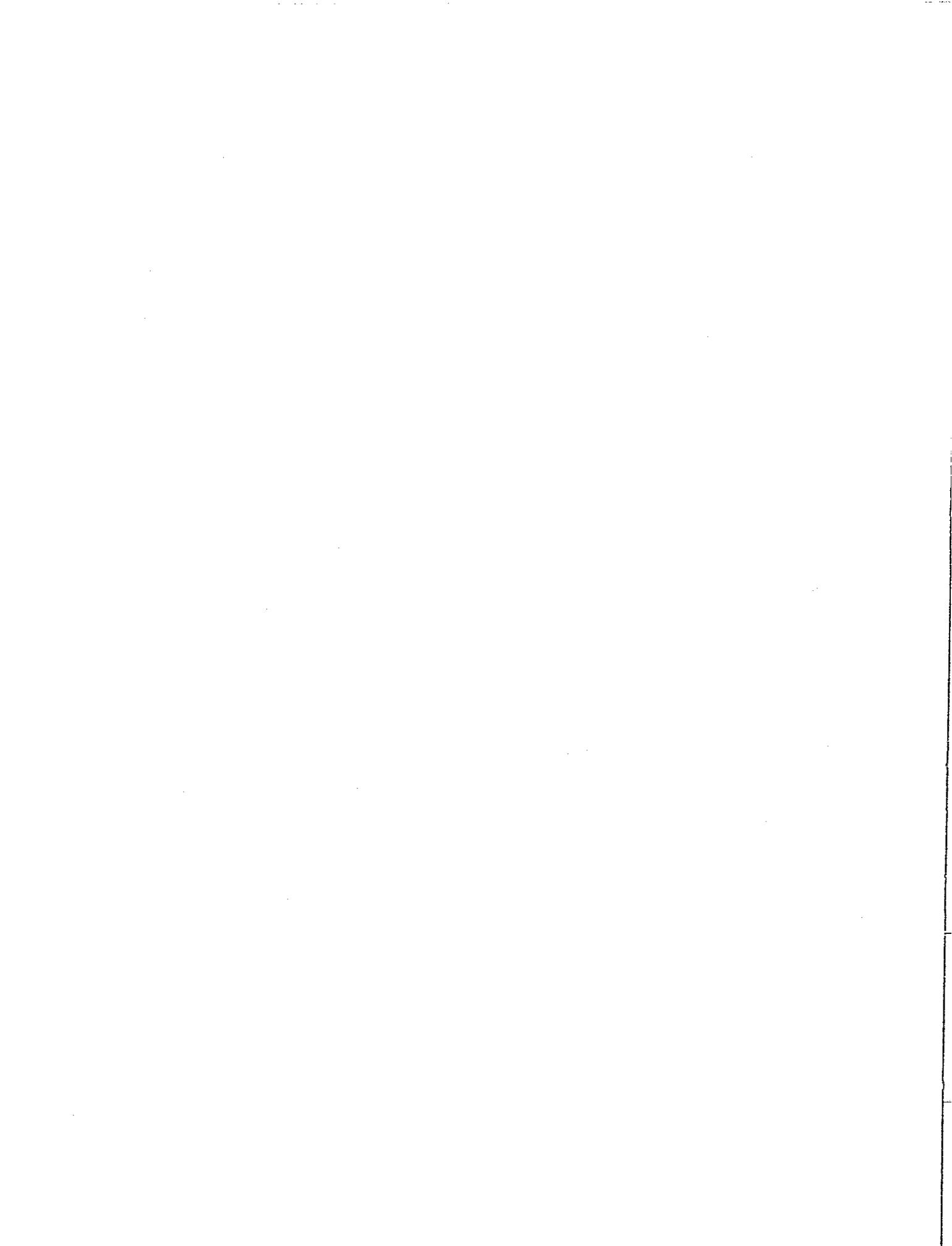
Uno de los resultados más interesantes es que tan bien una representación alternada como los ángulos de elevación sagital trabaja para la animación de los pasos. Hay otras representaciones del movimiento interesantes (de otra manera los ángulos de unión) que todavía serán descubiertos para otras tareas de animación.

Las restricciones usadas en la cadera en la unión compleja, controlando la lista pélvica, el dedo del pie afuera y la anchura de giro, fueron elegidas porque estas cantidades se relacionan más directamente a las uniones sin usar DOFs, eso es, la unión DOFs no involucran significativamente en satisfacer las restricciones del ángulo de elevación. La restricción de anchura de posición usada en la unión compleja de la posición del tobillo era escogida para su analogía a la restricción de anchura de giro.

Se usaron cinco conjuntos de datos generando todos los ejemplos, en todas las figuras. Se pueden usar pocos conjuntos de datos debido a la representación de la invariancia del ángulo de elevación sagital con respecto al tamaño de la figura y la habilidad de generar la curva de locomoción cambiando la dirección del plano sagital. Esto significó que no se necesitan los datos para cada curvatura y combinación de inclinación.

MetaGait modifica los parámetros de nivel altos, como la longitud, la altura del paso, dedo del pie afuera y encabezamiento de la dirección, basado en el terreno y la trayectoria que este está siguiendo. Hay más relaciones mutuas que no planeamos sin embargo. Por ejemplo, se modifican los parámetros de anchura de paso largo durante la curva de locomoción y no se tiene una relación explícita entre la trayectoria curva y la longitud del paso, curvas diferentes de altura / de longitud. Investigar en estas relaciones mutuas será crucial al modelado apropiado y realista de pasos.

El modelo sólo genera el movimiento para el cuerpo más bajo. Actualmente se genera el movimiento del cuerpo superior teniendo los brazos opuestos a la pierna para el brazo del giro y doblando la espina para compensar para el movimiento de la pelvis, para que todavía permanezca relativamente la cabeza. Sin embargo, para investigar el movimiento del cuerpo superior se necesita penosamente mejorar el realismo de la animación del caminado. La investigación puede extenderse a los movimientos para correr además de caminar. Esto requeriría la suma de alguna dinámica del modelo que se cree también podría mejorar la veracidad del modelo de interacción pie-suelo, incluso la suma de una fase dinámica de la posición doble. El modelo tampoco compensa para los objetos en la trayectoria del pie de giro; se necesita una investigación adicional en cómo modificar el movimiento del caminado para evitar los obstáculos bajos.



Conclusiones.

Este trabajo logra el objetivo de representar un caminar semejante al caminado de un humano real, sin embargo puede haber soluciones más adecuadas y otras más limitadas.

Para imitar el caminado humano se generó un modelo representado por un esqueleto jerárquico compuesto de segmentos conectados por ángulos y sirve para definir las partes del movimiento de la figura. Se realizó un análisis y diseño de los componentes del cuerpo humano para describir su movimiento. También un análisis, diseño y control del movimiento del caminado, empleando la relación lineal (el jacobiano) que mapea los cambios en los ángulos variables θ a cambios x en la posición y orientación en el resultado final, invirtiendo esta relación (el jacobiano inverso) provee las bases para controlar el movimiento. Aplicando así la cinemática en los brazos, piernas y pies del esqueleto logra cada una de sus metas realizando coordinaciones en el cuerpo que ayudan a describir la secuencia de poses que muestra los movimientos de cada paso necesarios para manifestar una animación: el movimiento del caminar humano.

El diseño cinemático corresponde a generar configuraciones apropiadas del movimiento de caminado que procesadas en un programa de animación generan la animación del caminado. La cinemática inversa se resuelve con los métodos iterativos y en general estos métodos son convenientes y bastantes favorables para la simulación de la cinemática. El sistema resuelve el problema de la generación del movimiento de pasos al usar cinemática para los movimientos generadores de pasos del caminado humano para aplicaciones en ambiente virtuales.

El algoritmo ha sido probado sobre una PC Pentium III a 600MHz, 256 MB en RAM trabajando en un ambiente Windows 98, en Visual C++ y de una interfaz Renderman que usa el programa llamado BMRT (Blue Moon Rendering Tools) para describir la escena del caminar de la figura articulada y mejorar la calidad gráfica de desplegado. Tiene un tiempo

de ejecución de 1 segundo en un ciclo de caminar (2 pasos, $\frac{1}{2}$ segundo por paso) El algoritmo da un nivel de desempeño esperado al comparar el caminar humano con el caminar de la figura articulada obtenida. También se compararon los movimientos generados contra aquellos movimientos de los humanos reales.

Algunos trabajos futuros son el considerar otras geometrias para representar el modelo humano, realizar una animación interactiva de esta figura articulada. Generar una herramienta automatizada para crear los pasos humanos. Hay otras representaciones del movimiento interesantes que todavía serán descubiertos para otras tareas de animación. Se puede extender la investigación a los movimientos para correr además de caminar, se necesita también una investigación adicional en cómo modificar el movimiento del caminado para evitar los obstáculos.

Trabajos que el agente virtual podría realizar, algunos de éstos son la tarea de reconocer y acomodar frascos, bailar, botar una pelota, etc. Las tendencias hacia la red de computadoras junto con sistemas gráficos y ambientes virtuales en años recientes levantó el interés por la investigación en el campo de ambientes virtuales distribuidos. Además da pauta a la investigación para buscar nuevas aplicaciones en diferentes áreas por mencionar algunas tales como Inteligencia Artificial, Redes Neuronales, Lógica Difusa, Lógica Borrosa, Trabajos Virtuales, Sistemas Interactivos, Terapia de Entrenamiento o Físico.

Las contribuciones de este trabajo son el análisis y la descripción del movimiento del humano, donde el diseño y control del caminar dependen de la coordinación del movimiento de cada una de las partes del cuerpo y de su control del movimiento al aplicar la cinemática inversa, ya que una parte afectará la posición de las otras partes. La restricción es una meta que la cinemática inversa intenta satisfacer para adoptar la mejor configuración, por lo tanto las restricciones definen las metas que se deben alcanzar en los brazos, piernas y pies para que las posturas del movimiento humano parezcan naturales. Las ecuaciones algebraicas para los brazos y piernas hacen posible el empleo de la cinemática y la generación de las poses para el caminado. La contribución principal es el análisis comparativo del cuerpo humano con el modelo de la figura articulada y así generar el movimiento para la parte baja del cuerpo y el movimiento para el cuerpo superior mejorando el realismo de la animación del caminado.

Por lo tanto una de las acciones más importantes que generan los pasos humanos es su animación del caminar.

Las configuraciones o posturas obtenidas dan las poses del movimiento humano en su acción que realizan al caminar.

ANEXO A.

DIAGRAMA DE BLOQUES.

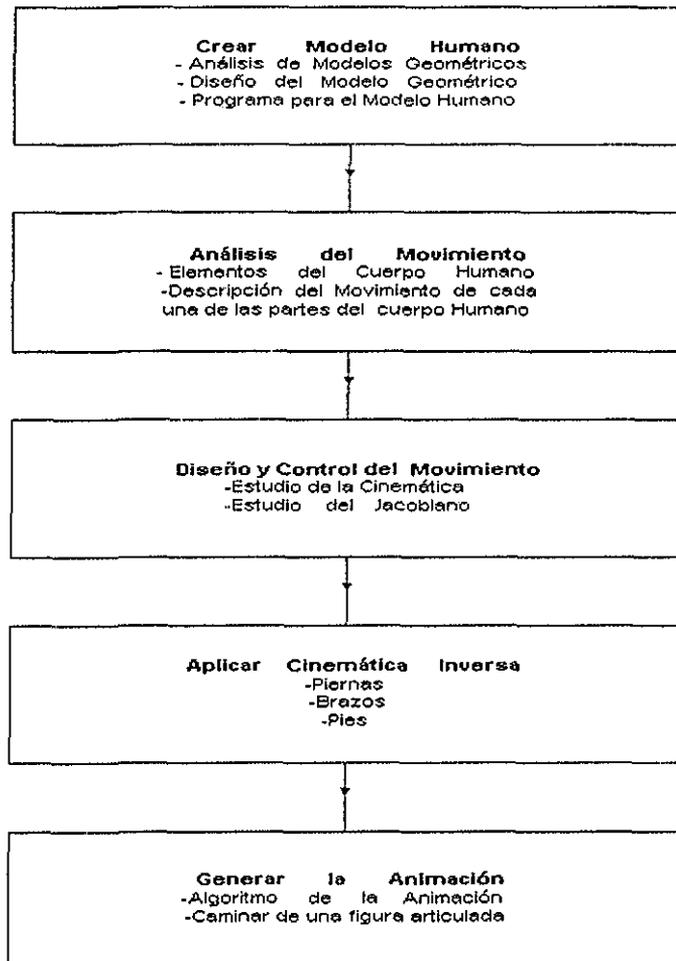
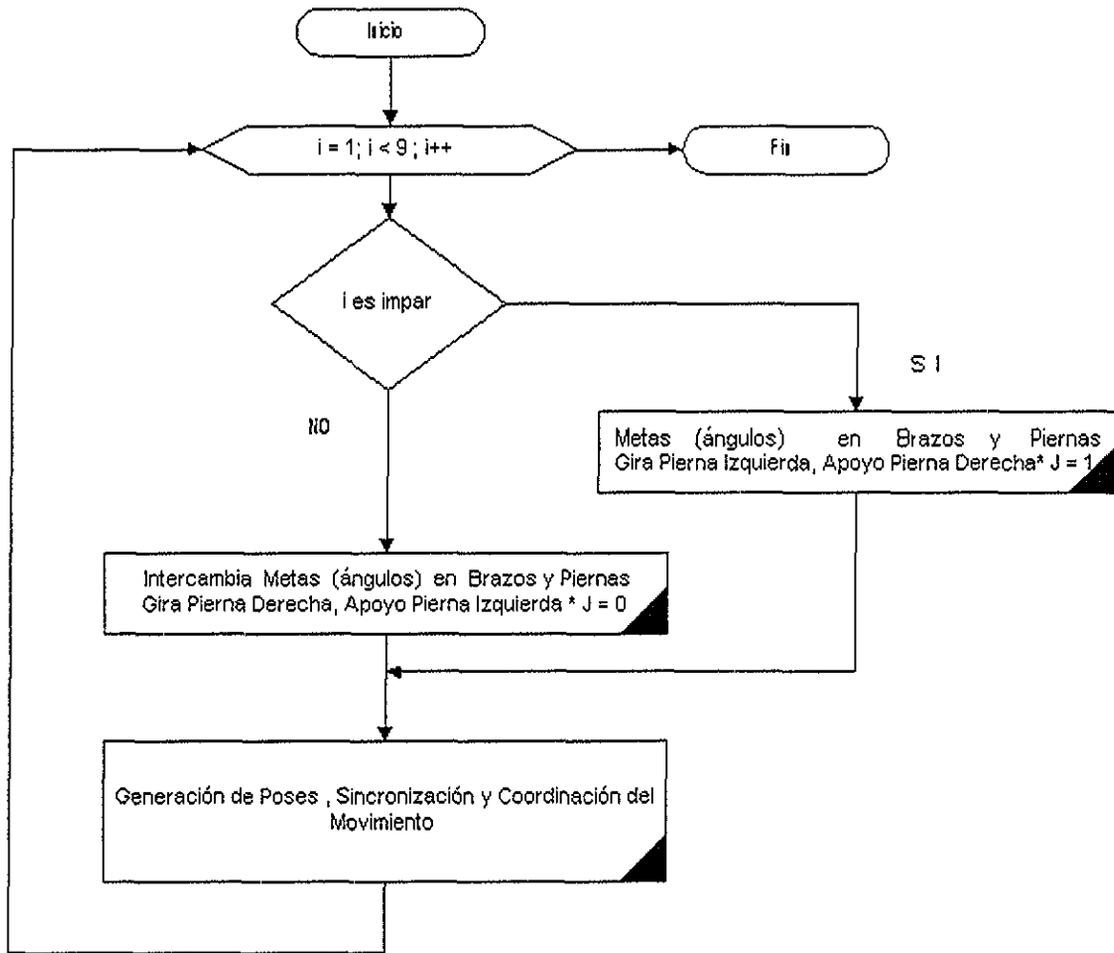


Figura A1. DIAGRAMA DE BLOQUES: ANIMACIÓN. EL CAMINAR DE UNA FIGURA ARTICULADA.

ANEXO B.

Diagrama de Flujo y Módulos del Algoritmo



* Esta en función de sus posiciones Iniciales

Figura B0. Programa principal para animar el caminar humano, genera 8 pasos.

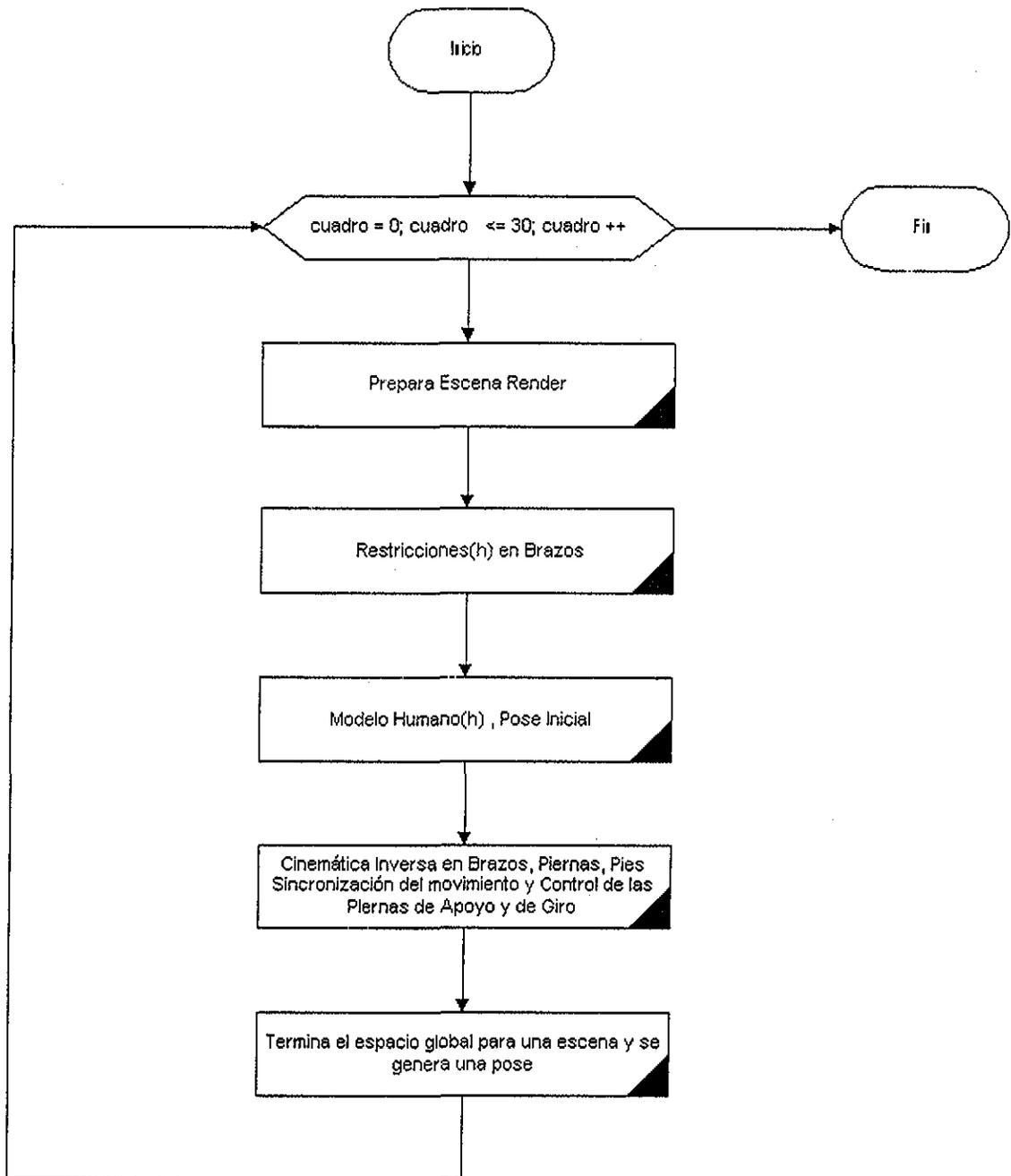


Figura B.0.1 Generación de Poses, Sincronización y Coordinación del Movimiento, para un paso.

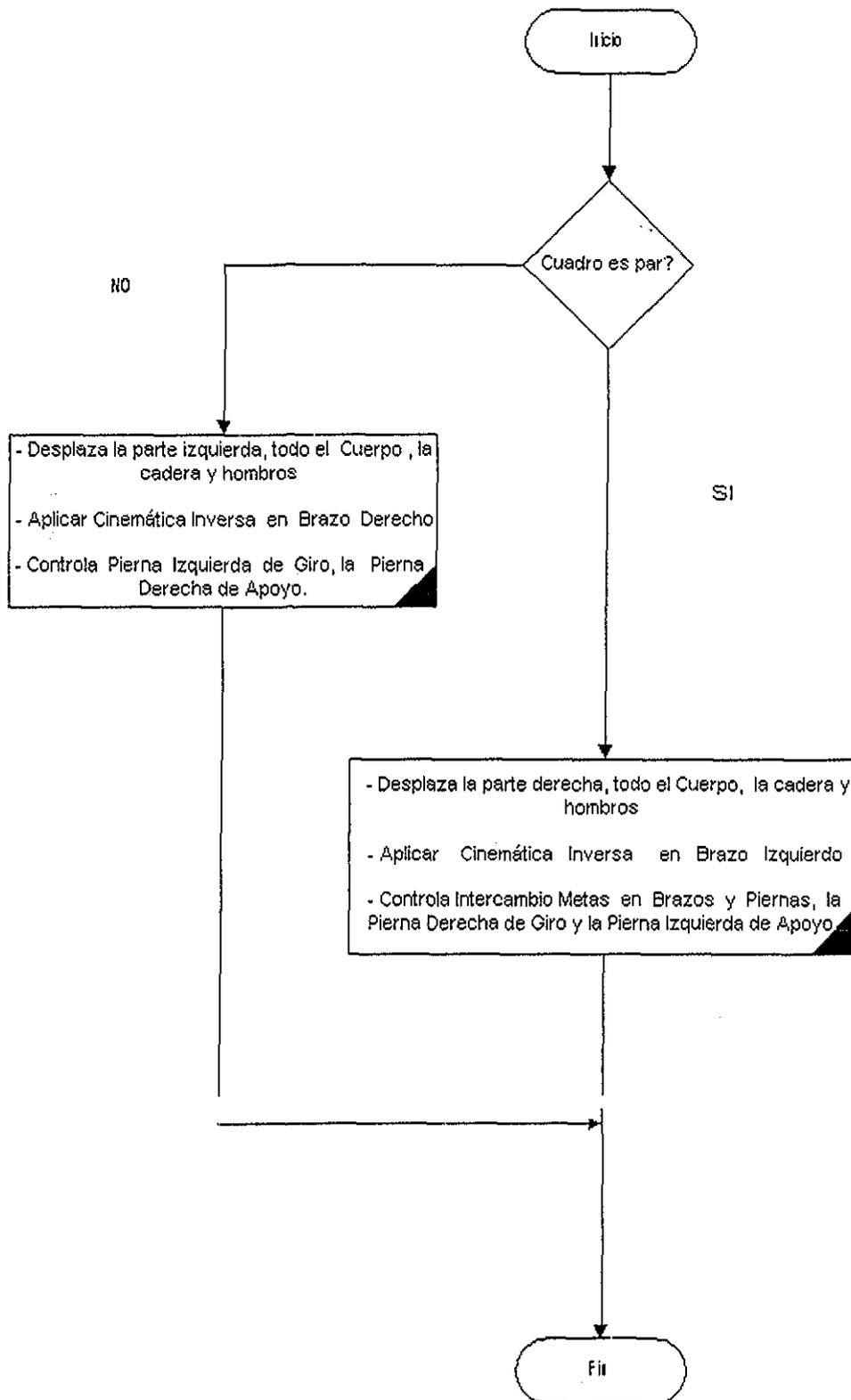


Figura B.0.2. Cinemática inversa en brazos, piernas y pies. Sincronización del movimiento y control de las piernas de apoyo y giro.

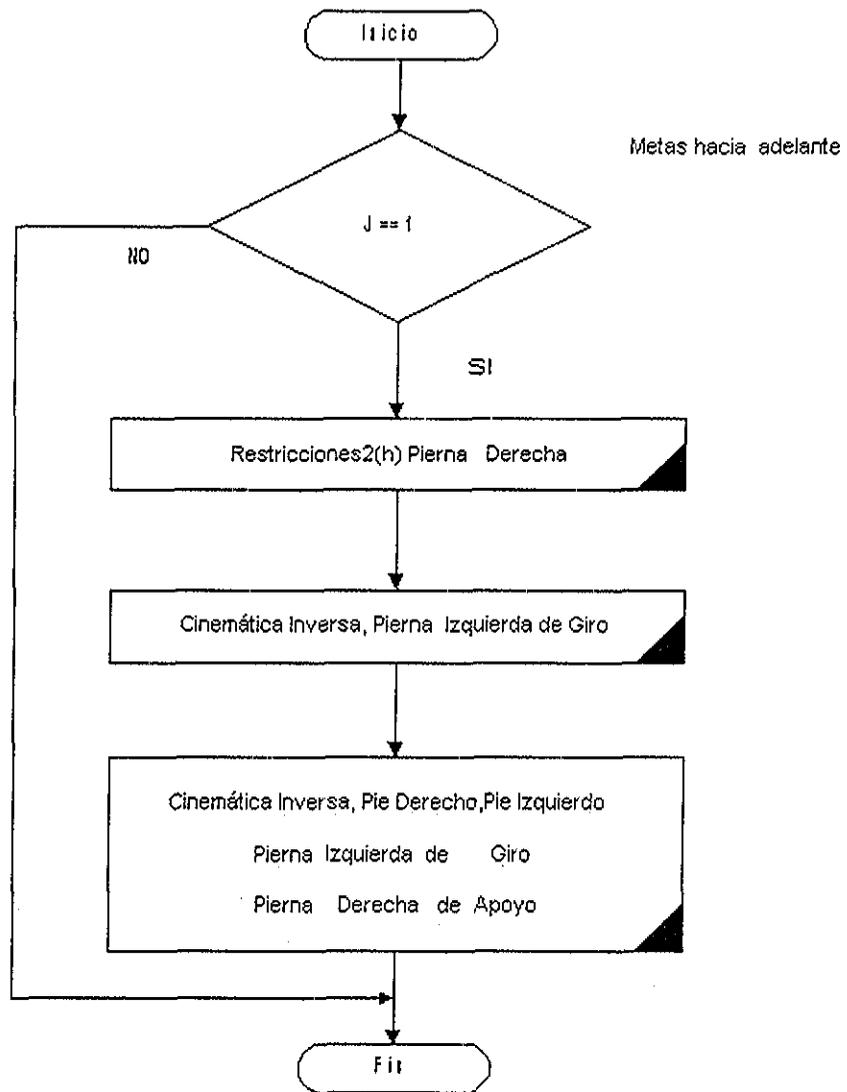


Figura B.0.3. Controla pierna izquierda de giro y la pierna derecha de apoyo.

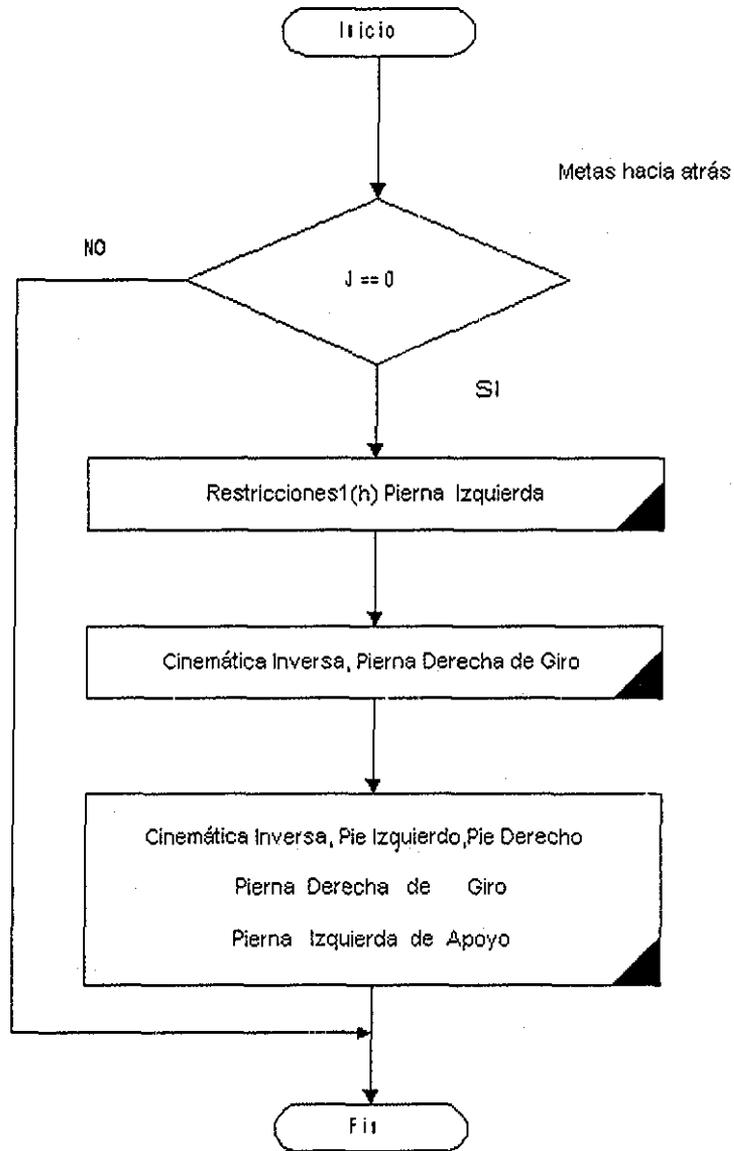


Figura B.0.4. Controla intercambio metas en brazos y piernas, la pierna derecha de giro y la pierna izquierda de apoyo.

Módulos del Algoritmo para Animar el Caminar.

- **Programa Principal. La animación** El caminar de una figura articulada generada por una secuencia de poses obtenidas empleando cinemática en las partes de los brazos, piernas y pies del cuerpo humano. Se tienen dos ciclos. El primero para controlar el número de pasos del caminar humano con el intercambio de metas en brazos y piernas, la pierna de giro y de apoyo en piernas y en los brazos sus posiciones. Las posiciones del brazo izquierdo adelante, la pierna derecha (apoyo) adelante, el brazo derecho atrás, la pierna izquierda(giro) atrás, son la configuración de inicio para iniciar el paso en la figura, en el intercambio del paso estas condiciones son puestas para el siguiente paso y estos pasos se repiten intercambiando sus configuraciones. El segundo para generar las poses, la sincronización y coordinación del movimiento para un paso del caminar aplicando la cinemática en brazos, piernas y pies, así todas las poses dan vida a la animación del caminar humano. El programa consta de las siguientes funciones:

```

// Esqueleto Representación de la Figura Humana
void ModeloHumano(humano H)
// Configuraciones Iniciales para una Pose en Particular de la Figura.
// Recibe como Parámetro el avance del Esqueleto.
humano humanoInit(RtFloat avance)
// Restricciones para los Brazos.
void Restricciones(humano H)
// Restricciones para la Pierna Izquierda.
void Restricciones1(humano H)
// Restricciones para la Pierna Derecha.
void Restricciones2(humano H)
// Cinemática Inversa para la Pierna Izquierda.
void AnimaPieIzquierdo(RtFloat x, RtFloat y, RtFloat z, humano hum)
// Cinemática Inversa para la Pierna Derecha.
void AnimaPieDerecho(RtFloat x, RtFloat y, RtFloat z, humano hum)
// Cinemática Inversa para el Brazo Izquierdo.
void AnimaArmIzquierdo(RtFloat x, RtFloat y, RtFloat z, humano hum)
// Cinemática Inversa para el Brazo Derecho.
void AnimaArmDerecho(RtFloat x, RtFloat y, RtFloat z, humano hum)
// Movimiento de la Cadera, Hombros, la Pierna Posterior y el Avance del
Esqueleto.
void MovimientoCadera(humano hum, bool izquierdo)

```

- **Generación de Poses, Sincronización y Coordinación del Movimiento para un paso.** Este módulo genera las poses de los movimientos realizados por cada una de las partes del cuerpo por ejemplo las piernas, los brazos, los pies, la cadera, los hombros, cabeza, etc., en un paso y consta de los siguientes módulos:

Prepara Escena Render. Se usan algunas rutinas de la interfaz Renderman para generar una escena de alta calidad o ambiente para el caminar. Para ello se construyen dos paredes y un piso que es el área del caminar de la figura articulada

empleando además color, luz y sombra de la figura en la escena, almacenando la pose obtenida en un archivo.

RiBegin(render) Comienza el render

RiFrameBegin() Comienza escena

Control de la cámara, almacenado de poses en archivo

RiDisplay(), RiProjection(), RiTranslate(), RiRotate(), Ricolor()

RiWorldBegin() Comienza espacio global

Genera paredes, piso, luz, figura articulada y sombra

**RiLightSource(), RiAttributeBegin(), RiDeclare(), RiSurface
(), RiPolygon(), RiTexture(), RiPatch(), RiAttributeEnd()**

Programa principal Genera poses para la animación

RiWorldEnd() Termina espacio global

RiFrameEnd() Termina escena

RiEnd() termina render

Restricciones. Alcance máximo en ángulos para Brazos. El módulo Restricciones(humano h) controla los ángulos que se consideran como límites en los brazos y puedan tener un comportamiento adecuado en su movimiento al caminar. Los ángulos son considerados de acuerdo a su eje coordenado y de esa manera son asignados sus valores. Para el brazo izquierdo se tiene:

XRotBrazo1 = -70, XRotBrazo1 = -60, xRotCodo1 = 0

YRotBrazo1 = -60, YROT Brazo1 = 90, xRotCodo1 = 170

Para el brazo derecho se tiene:

XRotBrazo1 = -70, XRotBrazo1 = -60, xRotCodo1 = 0

YRotBrazo1 = -60, YROT Brazo1 = 90, xRotCodo1 = 170

- Estos ángulos son elegidos con el fin de evitar inestabilidades, singularidades en el proceso de solución y de alguna forma de optimizar los movimientos en los brazos.

Modelo Humano. Hacer un humano sintético es una tarea compleja, incluso modelar el esqueleto. Para modelar el cuerpo humano se analizó y seleccionó una representación apropiada. Porque el cuerpo humano es muy complejo, se usó el modelo simplificado de un esqueleto. El cuerpo humano consiste de huesos y uniones que forman una estructura jerárquica - el árbol. El esqueleto puede representarse como un conjunto de objetos conectados con las uniones. La complejidad del esqueleto es dada por los grados de libertad (DOF) donde un DOF se representa por un eje rotatorio. Así que para conseguir una aproximación más detallada se requieren más de 200 DOF. El esqueleto empleado en este trabajo requirió de 22 DOF aproximadamente. El modelo simplificado del cuerpo humano se encuentra en la función Modelo Humano(h) en donde se da el modelo geométrico humano, partiendo de su estructura de datos (Typedef struct Carácter* Humano) variables que contienen sus ángulos y dimensiones de sus partes del cuerpo, considerando la geometría de cilindros y esferas para su creación apoyándose de transformaciones (translación y rotación).

Cinemática Inversa en Brazos, Piernas, Pies, Sincronización del Movimiento y Control de las Piernas de Apoyo y de Giro. Se calculan las cinemáticas en brazo derecho, pierna izquierda (giro) y pies en cuadros impares, en cuadros pares se calculan la cinemática en el brazo izquierdo y al intercambio de metas esto se realiza en forma opuesta. Los módulos que contiene este módulo son:

- **Desplaza la parte izquierda, todo del cuerpo, la cadera y hombros.**
El módulo **MovimientoCadera(h, true)** desplaza la pierna derecha, el brazo derecho y todo el cuerpo para sincronizar su movimiento. Finalmente se ajustan todos los movimientos al desplazar el hombro izquierdo, en forma opuesta la cadera derecha, quedando con esto la pose lista para aplicar la cinemática y con ello generar la pose del movimiento

El módulo **MovimientoCadera(h, false)** desplaza la pierna izquierda, el brazo izquierdo y todo el cuerpo para sincronizar su movimiento. Finalmente se ajustan todos los movimientos al desplazar el hombro derecho, en forma opuesta la cadera izquierda, quedando con esto la pose lista para aplicar la cinemática y con ello generar la pose del movimiento.

- **Cinemática Inversa en Brazos, obtiene ángulos de desplazamiento.**
Brazos. Los módulos **AnimaArmlzquierdo(d1,e1,f1,h)**, **AnimaArmDerecho(d2,e2,f2,h)** tienen como entrada la posición final o meta que se pretende alcanzar, en estos casos estos parámetros son $x_m = d1$, $y_m = e1$, $z_m = f1$, $x_m = d2$, $y_m = e2$, $z_m = f2$, la variable **h** contiene la estructura del esqueleto humano con sus variables, ángulos y dimensiones de su cuerpo, esta variable se usa como entrada y salida y es aquí donde contiene los ángulos de la posición inicial de la figura que se emplea en la cinemática directa para obtener sus posiciones iniciales de los brazos, con esta información se obtienen los ángulos de desplazamiento usando el algoritmo de la cinemática inversa descrita en la figura B1 y alcanzar la meta que se da en los parámetros $d1$, $e1$, $f1$ y $d2$, $e2$, $f2$. Lo que da el movimiento para cada uno de estos brazos.
- **Controla la parte Izquierda de Giro, la Pierna Derecha de Apoyo.** Si la posición inicial es $j == 1$ se tienen los módulos siguientes:

Restriciones2(humano H) para la pierna derecha, controlan los ángulos que se consideran como límites en las piernas y puedan tener un comportamiento adecuado en su movimiento al caminar. Los ángulos son considerados de acuerdo a su eje coordenado y de esa manera son asignados sus valores.

para la pierna derecha se tiene:

XRotCadera1 = -70, XRotCadera1 = -60, xRotRodilla1 = 0

YRotCadera1 = -60, YRotCadera1 = 90, xRotRodilla1 = 170

Cinemática Inversa en Piernas, obtiene ángulos de desplazamiento.

AnimaPieIzquierdo(a1,b1,c1,h), tienen como entrada la posición final o meta que se pretende alcanzar , en estos casos estos parámetros son $x_m=a1$, $y_m=b1$, $z_m=c1$, usando el algoritmo de la cinemática inversa descrita en la figura B1 y alcanzar la meta que se da en los parámetros $a1$, $b1$, $c1$ y $a2$, $b2$, $c2$. Lo que da el movimiento para esta pierna.

Cinemática Inversa aplicada a los Pies.

Del estudio del movimiento en los pies consideran que de los ángulos obtenidos en las piernas para la cadera y rodilla se obtiene el comportamiento de los pies, para aplicar estos criterios se considera lo siguiente para cada pie

Pie Izquierdo.

- Si el pie de izquierdo es el que gira se tiene lo siguiente para los valores de ángulos en los pies
 - PieIzquierdo($H \rightarrow x_{RotTobillo1}$) = $H \rightarrow x_{RotCadera1}$ + $H \rightarrow x_{RotRodilla1}$
 - PieDerecho($H \rightarrow x_{RotTobillo2}$) = 0, pues es el pie de apoyo.

- Controla intercambio metas en brazos y piernas, la pierna derecha de giro y la pierna izquierda de apoyo. Si la posición inicial es $j == 1$ se tienen los módulos siguientes:

Restricciones1(humano H) para la pierna izquierda

para la pierna izquierda se tiene:

$$\begin{aligned} X_{RotCadera1} &= -70, & X_{RotCadera1} &= -60, & x_{RotRodilla1} &= 0 \\ Y_{RotCadera1} &= -60, & Y_{RotCadera1} &= 90, & x_{RotRodilla1} &= 170 \end{aligned}$$

AnimaPieDerecho(a2,b2,c2,h) tienen como entrada la posición final o meta que se pretende alcanzar , en estos casos estos parámetros son $x_m=a2$, $y_m=b2$, $z_m=c2$, con esta información se obtienen los ángulos de desplazamiento usando el algoritmo de la cinemática inversa descrita en la figura B1 y alcanzar la meta que se da en los parámetros $a1$, $b1$, $c1$ y $a2$, $b2$, $c2$ Lo que da el movimiento para esta pierna

Cinemática Inversa aplicada a los Pies.

Pie Derecho.

Si el pie derecho es el que gira se tiene lo siguiente para los valores de ángulos en los pies

$$\begin{aligned} PieDerecho(H \rightarrow x_{RotTobillo2}) &= H \rightarrow x_{RotCadera2} + H \rightarrow x_{RotRodilla2} \\ PieIzquierdo(H \rightarrow x_{RotTobillo1}) &= 0, \text{ pues es el pie de apoyo} \end{aligned}$$

Módulo Cinemática Inversa.

1. **Dimensiones (tamaño en unidades)** de pierna, columna, espalda, pelvis, contenida en la estructura del esqueleto humano
2. **Valores iniciales** $\theta_1, \theta_2, \theta_3$ (por ejemplo: $\theta_1 = x_{cadera1}$, $\theta_2 = y_{cadera1}$, $\theta_3 = x_{rodilla1}$ en las partes que componen las uniones de la pierna izquierda), en brazos y piernas
3. Evalúa estos valores en la **expresión algebraica para brazos y piernas** ver sección 5.6 y **obtiene la posición x_0, y_0, z_0** inicial, emplea cinemática directa.

4. **Construye el Jacobiano** $J = \begin{bmatrix} \partial X_0 / \partial \theta_1 \\ \partial Y_0 / \partial \theta_2 \\ \partial Z_0 / \partial \theta_3 \end{bmatrix}$

5. **Encuentra una delta pequeña o cambio diferencial** $dx = X_m - X_0/n$, $dy = Y_m - Y_0/n$, $dz = Z_m - Z_0/n$, n es considerada como $n = 0.1$

6. **Usa la regla de cramer y calculo el Jacobiano inverso**

$$J^{-1} = \text{adj}(J) / \det(J)$$

7. **Cálculo determinante $\det(J)$** Si es cero, no cambian los ángulos y $\det(J) = 1.0$

8. **Cálculo el desplazamiento de sus ángulos.** $d\theta_i = [d\theta_1][d\theta_2][d\theta_3]$

$$d\theta_i = J^{-1} * \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix}$$

9. **Calcula el cambio en los ángulos θ_i [$\theta_1, \theta_2, \theta_3$]** dando el **desplazamiento real** de ellos, por ejemplo en las uniones de la pierna izquierda se tiene:

$$X_{\text{RotCadera1}} + = d\theta_1$$

$$Y_{\text{RotCadera1}} + = d\theta_2$$

$$Z_{\text{rotRodilla1}} + = d\theta_3$$

Y en forma general sería: $\theta_i = \theta_i + d\theta_i$, ver capítulo 5.

Figura B.1 Algoritmo de la Cinemática Inversa. Aplicada en Brazos y Piernas, ver su código en el anexo C.

ANEXO C.

PROGRAMA DE ANIMACIÓN

/*
UNAM.
Maestría en Ciencia e Ingeniería de la Computación.

Directora de Tesis: Mat. Ana Luisa Solis Gonzalez Cosio

Alumno : Ricardo Duarte Pérez

Programa que Realiza Un Modelo para el Control del Movimiento
Humano basado en Cinemática Inversa, generando la Animación
de Caminado de una Figura Articulada.

*/

```
#include <stdlib.h>  
#include <stdio.h>  
#include <string.h>  
#include <math.h>  
#include <ri.h>
```

```
// Estructura de Datos para la Figura Articulada (Esqueleto).
```

```
struct Caracter {  
    // Rotaciones, ángulos de rotación  
    RtFloat xRotBrazo1;    // Brazo Izquierdo en x, y, z color azul  
    RtFloat yRotBrazo1;  
    RtFloat zRotBrazo1;  
    RtFloat xRotCodo1;    // Codo Izquierdo en x color azul  
    RtFloat xRotBrazo2;    // Brazo Derecho en x, y, z  
    RtFloat yRotBrazo2;  
    RtFloat zRotBrazo2;  
    RtFloat xRotCodo2;    // Codo Derecho en x color rojo
```

```

RtFloat xRotRodilla1; // Rodilla Izquierda en x color azul
RtFloat xRotCadera1; // Cadera Izquierda en x, y, z color azul
RtFloat yRotCadera1;
RtFloat zRotCadera1;

RtFloat xRotRodilla2; // Rodilla Derecha en x color rojo
RtFloat xRotCadera2; // Cadera Derecha en x, y, z color rojo

RtFloat yRotCadera2;
RtFloat zRotCadera2;

RtFloat RotPelvis; // Pelvis.
RtFloat Avance; // Desplazamiento del Esqueleto.

RtFloat xRotTobillo1; // Tobillos en x.
RtFloat xRotTobillo2;
RtFloat xRotPuntaPie; // Puntas de los Pies en x.
RtFloat xRotCuello; // Cuello en x, y, z.
RtFloat yRotCuello;
RtFloat xRotEspalda; //Espalda en x, y, z.
RtFloat yRotEspalda;
RtFloat zRotEspalda;
    // Longitudes
RtFloat RadioArticulacion;
RtFloat RadioHueso;
RtFloat rCabeza;
RtFloat lColumna;
RtFloat wPelvis;
RtFloat lBrazo;
RtFloat lPierna;
RtFloat wEspalda;
RtFloat lPie;
RtFloat k;
};

typedef struct Caracter* humano;

RtPoint piso [4] = {
    {-120,-156,500},{-120,-160,-400},{0,-156,500},{0,-160,-400}};
RtPoint pfte [4] = {
    {-120,-156,500},{-120,-160,-400},{-120,200,500},{-120,200,-400}};
RtPoint plado [4] = {
    {-120,-160,-400},{0,-160,-400},{0,200,-400},{-120,200,-400}};

```

```

void ModeloHumano(humano H)
{
    RtColor color;          // Color Azul del Esqueleto.

    color[0] = 0.0f;
    color[1] = 0.0f;
    color[2] = 1.0f;
    RiColor(color);

    RiTransformBegin();
    RiScale(1.5,1.5,1.5);
    RiTranslate(0.0f, 10.0f, -100.0);
    RiTranslate(0.0f, 0.0f, H->Avance); //Avanza el Esqueleto.

    // Torso (Espalda, Columna) y Pelvis.
    color[0] = 1.0f; // Color Rojo
    color[1] = 0.0f;
    color[2] = 0.0f;
    RiColor(color);
    RiRotate(90.0f, 0.0f, 1.0f, 0.0f);
    RiTranslate(0.0, -H->lColumna, -H->wPelvis/2);
    RiRotate(H->RotPelvis, 0.0f, 1.0f, 0.0f);
    RiCylinder(H->RadioHueso, 0, H->wPelvis, 360.0,RI_NULL);
    color[0] = 0.0f;
    color[1] = 1.0f;
    color[2] = 0.0f;
    RiColor(color);
    RiTranslate(0, 0, H->wPelvis/2);
    RiRotate(-90, 1.0, 0.0, 0.0);
    RiRotate(H->yRotEspalda/2 -2, 1.0f, 0.0f, 0.0f);
    RiRotate(-H->xRotEspalda/2 -2, 0.0f, 1.0f, 0.0f);
    RiRotate(H->zRotEspalda/2 -2, 0.0f, 0.0f, 1.0f);
    RiCylinder(H->RadioHueso, 0, H->lColumna/2, 360.0,RI_NULL);
    RiTranslate(0.0, 0.0, H->lColumna/2);
    RiRotate(H->yRotEspalda/2 -2, 1.0f, 0.0f, 0.0f);
    RiRotate(-H->xRotEspalda/2 -2, 0.0f, 1.0f, 0.0f);
    RiRotate(H->zRotEspalda/2 -2, 0.0f, 0.0f, 1.0f);
    RiCylinder(H->RadioHueso, 0, H->lColumna/2 + 5, 360.0,RI_NULL);
    RiSphere(H->RadioArticulacion, -H->RadioArticulacion, H->RadioArticulacion,
    360.0,RI_NULL);
    RiTranslate(0.0, 0.0, -H->lColumna/2);
    RiRotate(-90.0f, 0.0f, 1.0f, 0.0f);
    RiRotate(-90.0f, 0.0f, 0.0f, 1.0f);
    RiTranslate(-0.0, H->lColumna + 5, 0.0);

```

```

color[0] = 0.0f; // Color Azul
color[1] = 0.0f;
color[2] = 1.0f;

```

```

RiColor(color); // Cabeza y Cuello

```

```

RiTransformBegin();
RiTranslate(0.0f, H->rCabeza + 5, 0.0f);
RiCylinder(H->RadioHueso, 0, H->lColumna/3, 360.0,RI_NULL);
RiSphere(H->rCabeza, -H->rCabeza, H->rCabeza, 360.0,RI_NULL);
RiTranslate(0.0,-H->rCabeza-5,0.0);
RiRotate(-90.0f, 1.0f, 0.0f, 0.0f);
RiCylinder(H->RadioHueso, 0, H->rCabeza/2, 360.0,RI_NULL);
RiTransformEnd();

```

```

// Pierna Izquierda.

```

```

RiTransformBegin();
RiTranslate(-H->wPelvis/2, -H->lColumna - 5, 0);
RiRotate(90, 1.0f, 0.0f, 0.0f);
RiSphere(H->RadioArticulacion, -H->RadioArticulacion, H->RadioArticulacion,
360.0,RI_NULL);
RiRotate(H->yRotCadera1, 0.0f, 0.0f, 1.0f);
RiRotate(H->xRotCadera1, -1.0f, 0.0f, 0.0f);
RiRotate(H->zRotCadera1, 0.0f, 1.0f, 0.0f);
RiCylinder(H->RadioHueso, 0, H->lPierna/2, 360.0,RI_NULL);
RiTranslate(0.0f, 0.0f, H->lPierna/2);
RiSphere(H->RadioArticulacion, -H->RadioArticulacion, H->RadioArticulacion,
360.0,RI_NULL);
RiRotate(H->xRotRodilla1, -1.0f, 0.0f, 0.0f);
RiCylinder(H->RadioHueso, 0, H->lPierna/2, 360.0,RI_NULL);
RiTranslate(0.0f, 0.0f, H->lPierna/2);
RiRotate(90, -1.0, 0.0, 0.0);
RiRotate(-H->xRotTobillo1, -1.0f, 0.0f, 0.0f);
RiSphere(H->RadioArticulacion, -H->RadioArticulacion, H->RadioArticulacion,
360.0,RI_NULL);
RiCylinder(H->RadioHueso, 0, H->lPie, 360.0,RI_NULL);
RiTranslate(0.0f, 0.0f, H->lPie);
RiRotate(H->xRotPuntaPie, -1.0f, 0.0f, 0.0f);

```

```

RiSphere(H->RadioArticulacion, -H->RadioArticulacion, H->RadioArticulacion,
360.0,RI_NULL);
RiTransformEnd();

```

// Pierna Derecha.

```

RiTransformBegin();
color[0] = 1.0f; // Color Rojo
color[1] = 0.0f;
color[2] = 0.0f;
RiColor(color);
RiTranslate(H->wPelvis/2, -H->lColumna - 5, 0);
RiRotate(90, 1.0f, 0.0f, 0.0f);
RiSphere(H->RadioArticulacion, -H->RadioArticulacion, H->RadioArticulacion,
360.0,RI_NULL);
RiRotate(-H->yRotCadera2, 0.0f, 0.0f, 1.0f);
RiRotate(H->xRotCadera2, -1.0f, 0.0f, 0.0f);
RiRotate(-H->zRotCadera2, 0.0f, 1.0f, 0.0f);
RiCylinder(H->RadioHueso, 0, H->lPierna/2, 360.0,RI_NULL);
RiTranslate(0.0f, 0.0f, H->lPierna/2);
RiSphere(H->RadioArticulacion, -H->RadioArticulacion, H->RadioArticulacion,
360.0,RI_NULL);
RiRotate(H->xRotRodilla2, -1.0f, 0.0f, 0.0f);
RiCylinder(H->RadioHueso, 0, H->lPierna/2, 360.0,RI_NULL);
RiTranslate(0.0f, 0.0f, H->lPierna/2);
RiRotate(90, -1.0, 0.0, 0.0);
RiRotate(-H->xRotTobillo2, -1.0f, 0.0f, 0.0f);
RiSphere(H->RadioArticulacion, -H->RadioArticulacion, H->RadioArticulacion,
360.0,RI_NULL);
RiCylinder(H->RadioHueso, 0, H->lPie, 360.0,RI_NULL);
RiTranslate(0, 0, H->lPie);
RiRotate(H->xRotPuntaPie, -1.0f, 0.0f, 0.0f);
RiSphere(H->RadioArticulacion, -H->RadioArticulacion, H->RadioArticulacion,
360.0,RI_NULL);
RiTransformEnd();

```

// Hombros y Espalda.

```

RiTransformBegin();
RiRotate(90.0f, 0.0f, 1.0, 0.0f);
RiTranslate(0.0, 0.0, -H->wEspalda/2);
RiRotate(-H->RotPelvis, 0.0f, 1.0f, 0.0f);
RiCylinder(H->RadioHueso, 0, H->wEspalda, 360.0,RI_NULL);
RiTransformEnd();

```

// Brazo Izquierdo.

```

RiTransformBegin();
color[0] = 0.0f; // Color Azul
color[1] = 0.0f;
color[2] = 1.0f;
RiColor(color);
RiRotate(-H->RotPelvis, 0.0f, 1.0f, 0.0f);
RiRotate(90, 1.0, 0.0, 0.0);
RiTranslate(-H->wEspalda/2, 0.0, 0.0);
RiRotate(H->yRotBrazo1, 0.0f, 0.0f, 1.0f);

```

```

RiRotate(H->xRotBrazo1, -1.0f, 0.0f, 0.0f);
RiRotate(H->zRotBrazo1, 0.0f, 1.0f, 0.0f);
RiCylinder(H->RadioHueso, 0, H->IBrazo/2, 360.0,RI_NULL);
RiSphere(H->RadioArticulacion, -H->RadioArticulacion, H->RadioArticulacion,
360.0,RI_NULL);
RiTranslate(0.0, 0.0, H->wEspalda/2);
RiRotate(H->xRotCodo1, -1.0, 0.0, 0.0);
RiCylinder(H->RadioHueso, 0, H->IBrazo/2, 360.0,RI_NULL);
RiSphere(H->RadioArticulacion, -H->RadioArticulacion, H->RadioArticulacion,
360.0,RI_NULL);
RiTranslate(0.0, 0.0, H->wEspalda/2);
RiSphere(H->RadioArticulacion, -H->RadioArticulacion, H->RadioArticulacion,
360.0,RI_NULL);
RiTransformEnd();

```

// Brazo Derecho.

```

RiTransformBegin();
color[0] = 1.0f;      // Color Rojo
color[1] = 0.0f;
color[2] = 0.0f;
RiColor(color);
RiRotate(-H->RotPelvis, 0.0f, 1.0f, 0.0f);
RiRotate(90, 1.0, 0.0, 0.0);
RiTranslate(H->wEspalda/2,0.0,0.0);
RiRotate(-H->yRotBrazo2, 0.0f, 0.0f, 1.0f);
RiRotate(H->xRotBrazo2, -1.0f, 0.0f, 0.0f);
RiRotate(-H->zRotBrazo2, 0.0f, 1.0f, 0.0f);
RiCylinder(H->RadioHueso, 0, H->IBrazo/2, 360.0,RI_NULL);
RiSphere(H->RadioArticulacion, -H->RadioArticulacion, H->RadioArticulacion,
360.0,RI_NULL);
RiTranslate(0.0, 0.0, H->wEspalda/2);
RiRotate(H->xRotCodo2, -1.0, 0.0, 0.0);
RiCylinder(H->RadioHueso, 0, H->IBrazo/2, 360.0,RI_NULL);
RiSphere(H->RadioArticulacion, -H->RadioArticulacion, H->RadioArticulacion,
360.0,RI_NULL);
RiTranslate(0.0, 0.0, H->wEspalda/2);
RiSphere(H->RadioArticulacion, -H->RadioArticulacion, H->RadioArticulacion,
360.0,RI_NULL);
RiTransformEnd();
RiTransformEnd();
}

```

// Configuraciones Iniciales para una Pose en Particular de la Figura.
// Recibe como Parámetro el avance del Esqueleto.

```
humano humanoInit(RtFloat avance)
{
```

```
    // Ángulos de Rotación para las partes del Esqueleto //(Modelo Humano)
```

```
    humano h = (humano) malloc(sizeof *h);
```

```
    h->xRotBrazo1 = -14.61; // Brazo Izquierdo azul
```

```
    h->yRotBrazo1 = 0.0;
```

```
    h->zRotBrazo1 = 0.0;
```

```
    h->xRotCodo1 = 89.66; // Codo Izquierdo azul
```

```
    h->xRotBrazo2 = -13.93; // Brazo Derecho rojo
```

```
    h->yRotBrazo2 = 0.0;
```

```
    h->zRotBrazo2 = 0.0;
```

```
    h->xRotCodo2 = 10.60; // Codo Derecho rojo
```

```
    h->xRotCadera1 = -33.0; // Pierna Izquierda azul
```

```
    h->yRotCadera1 = 0.0;
```

```
    h->zRotCadera1 = 0.0;
```

```
    h->xRotRodilla1 = -29.0; // Rodilla Izquierda azul
```

```
    h->xRotRodilla2 = -29.0; // Rodilla Derecha rojo
```

```
    h->xRotCadera2 = 33.0; // Pierna Derecha rojo
```

```
    h->yRotCadera2 = 0.0;
```

```
    h->zRotCadera2 = 0.0;
```

```
    h->RotPelvis = -3.0;
```

```
    h->Avance = avance;
```

```
    h->xRotTobillo1 = 0.0;
```

```
    h->xRotTobillo2 = 0.0;
```

```
    h->xRotPuntaPie = 0.0;
```

```
    h->xRotCuello = 0.0;
```

```
    h->yRotCuello = 0.0;
```

```
    h->xRotEspalda = 0.0;
```

```
    h->yRotEspalda = 0.0;
```

```
    h->zRotEspalda = 0.0;
```

```
    h->RadioArticulacion = 2.0;
```

```
    h->RadioHueso = 1.0;
```

```
    h->rCabeza = 10.0;
```

```
    h->wPelvis = 20.0;
```

```
    h->lColumna = 40.0;
```

```
    h->wEspalda = 40.0;
```

```
    h->lBrazo = 40.0;
```

```
    h->lPierna = 60.0;
```

```

h->IPie      = 10.0;
h->k         = 0.0;
return h;
}

```

// Restricciones para los Brazos.

```
void Restricciones(humano H)
```

```

{
if (H->xRotBrazo1 < -70) H->xRotBrazo1 = -70;
if (H->xRotBrazo1 > 180) H->xRotBrazo1 = 180;
if (H->yRotBrazo1 < -60) H->yRotBrazo1 = -60;
if (H->yRotBrazo1 > 90) H->yRotBrazo1 = 90;
if (H->xRotCodo1 < 0) H->xRotCodo1 = 0;
if (H->xRotCodo1 > 170) H->xRotCodo1 = 170;
if (H->xRotBrazo2 < -70) H->xRotBrazo2 = -70;
if (H->xRotBrazo2 > 180) H->xRotBrazo2 = 180;
if (H->yRotBrazo2 < -60) H->yRotBrazo2 = -60;
if (H->yRotBrazo2 > 90) H->yRotBrazo2 = 90;
if (H->xRotCodo2 < 0) H->xRotCodo2 = 0;
if (H->xRotCodo2 > 170) H->xRotCodo2 = 170;
}

```

// Restricciones para la Pierna Izquierda.

```
void Restricciones1(humano H)
```

```

{
if (H->xRotCadera1 < -20) H->xRotCadera1 = -20;
if (H->xRotCadera1 > 20) H->xRotCadera1 = 20;
if (H->yRotCadera1 < -60) H->yRotCadera1 = -60;
if (H->yRotCadera1 > 120) H->yRotCadera1 = 120;
if (H->xRotRodilla1 < -20) H->xRotRodilla1 = -20;
if (H->xRotRodilla1 > 20) H->xRotRodilla1 = 20;
}

```

// Restricciones para la Pierna Derecha.

```
void Restricciones2(humano H)
```

```
{
    if (H->xRotCadera2 < -20) H->xRotCadera2 = -20;
    if (H->xRotCadera2 > 20) H->xRotCadera2 = 20;
    if (H->yRotCadera2 < -60) H->yRotCadera2 = -60;
    if (H->yRotCadera2 > 120) H->yRotCadera2 = 120;
    if (H->xRotRodilla2 < -20) H->xRotRodilla2 = -20;
    if (H->xRotRodilla2 > 20) H->xRotRodilla2 = 20;
}
```

```
// Cinemática Inversa para la Pierna Izquierda.
```

```
void AnimaPieIzquierdo(RtFloat x, RtFloat y, RtFloat z, humano hum)
```

```
{
    RtFloat x0, y0, z0, x1, y1, z1, det;
    RtFloat theta1, theta2, theta3, dtheta1, dtheta2, dtheta3;
    RtFloat lPierna, lColumna, wPelvis, wEspalda, lColumna1, lColumna2;
```

```
RtFloat matrix[3][3], Imatrix[3][3]; //Jacobiano y su Inversa.
```

```
lPierna = hum->lPierna/2;
lColumna1 = hum->lColumna/2;
lColumna2 = hum->lColumna/2 + 5;
lColumna = hum->lColumna + 5;
wEspalda = hum->>wEspalda;
wPelvis = hum->>wPelvis;
```

```
theta1 = hum->xRotCadera1/57.295f;
theta2 = hum->yRotCadera1/57.295f;
theta3 = hum->xRotRodilla1/57.295f;
```

```
// Encuentra la Posición desde los ángulos actuales;
// el pivote del tronco es (0, -30, 0)
```

```
x0 = - wPelvis/2 - (lPierna*(float)sin(theta3 + theta1) +
lPierna*(float)sin(theta1))*(float)sin(theta2);
```

```
y0 = -30 - (lPierna*(float)cos(theta3 + theta1) + lPierna*(float)cos(theta1));
```

```
z0 = (lPierna*(float)sin(theta3 + theta1) +
lPierna*(float)sin(theta1))*(float)cos(theta2);
```

```

// forma el jacobiano [J] =
// [JaJb Jc] <- [d(x0)/d(theta1) d(x0)/d(theta2) d(x0)/d(theta3)]
// [JdJd Jf]   [d(y0)/d(theta1) d(y0)/d(theta2) d(y0)/d(theta3)]
// [JgJh Ji]   [d(z0)/d(theta1) d(z0)/d(theta2)d(z0)/d(theta3)]

matrix[0][0] = -(lPierna*(float)cos(theta3+theta1) +
                lPierna*(float)cos(theta1))*(float)sin(theta2);

matrix[0][1] = -(lPierna*(float)sin(theta3+theta1) +
                lPierna*(float)sin(theta1))*(float)cos(theta2);

matrix[0][2] =      -lPierna*(float)cos(theta3+theta1)*(float)sin(theta2);

matrix[1][0] = -(-lPierna*(float)sin(theta3+theta1) - lPierna*(float)sin(theta1));

matrix[1][1] =      0;

matrix[1][2] =      lPierna*(float)sin(theta3+theta1);

matrix[2][0] =      (lPierna*(float)cos(theta3+theta1) +
                lPierna*(float)cos(theta1))*(float)cos(theta2);

matrix[2][1] =      -(lPierna*(float)sin(theta3+theta1) +
                lPierna*(float)sin(theta1))*(float)sin(theta2);

matrix[2][2] =      (lPierna*(float)cos(theta3+theta1))*(float)cos(theta2);

//
// Encuentra una Delta Pequeña r =
//
//                               [deltax/n]
//                               [deltay/n]
//                               [deltaz/n]

x1 = (x-x0)/0.1f;
y1 = (y-y0)/0.1f;
z1 = (z-z0)/0.1f;

// Resuelve la Inversa de la Matriz
// Usando la Regla de Cramer:
//  $J^{-1} = \text{adj}(J) / \det(J)$ 
// Calcula el Determinante de [J]
// Si det = 0, no Existe Inversa.

```

```

det = (
    + matrix[0][0]*(matrix[1][1]*matrix[2][2] - matrix[2][1]*matrix[1][2])
    - matrix[0][1]*(matrix[1][0]*matrix[2][2] - matrix[2][0]*matrix[1][2])
    + matrix[0][2]*(matrix[1][0]*matrix[2][1] - matrix[2][0]*matrix[1][1])
);

    // no puede Dividir entre 0;
    // si det == 0, no Existe Inversa ->
    // no Cambian los ángulos.

if (det == 0) det = 1.0f;

    // Inversa de [J] es (1/det)*adj(J)
Imatrix[0][0] = (matrix[1][1]*matrix[2][2] - matrix[1][2]*matrix[2][1])/det;
Imatrix[0][1] = -(matrix[0][1]*matrix[2][2] - matrix[0][2]*matrix[2][1])/det;
Imatrix[0][2] = (matrix[0][1]*matrix[1][2] - matrix[0][2]*matrix[1][1])/det;

Imatrix[1][0] = -(matrix[1][0]*matrix[2][2] - matrix[1][2]*matrix[2][0])/det;
Imatrix[1][1] = (matrix[0][0]*matrix[2][2] - matrix[0][2]*matrix[2][0])/det;
Imatrix[1][2] = -(matrix[0][0]*matrix[1][2] - matrix[0][2]*matrix[1][0])/det;

Imatrix[2][0] = (matrix[1][0]*matrix[2][1] - matrix[1][1]*matrix[2][0])/det;
Imatrix[2][1] = -(matrix[0][0]*matrix[2][1] - matrix[0][1]*matrix[2][0])/det;
Imatrix[2][2] = (matrix[0][0]*matrix[1][1] - matrix[0][1]*matrix[1][0])/det;

    //calcula dtheta = J-1 * r
    // [   ]{deltax/n} {dtheta1}
    // [ J-1 ]{deltay/n} = {dtheta2}
    // [   ]{deltaz/n} {dtheta3}

    // Cambio en los ángulos.
    // Entonces nueva_theta = theta + dtheta

dtheta1 = (Imatrix[0][0]*(x1) + Imatrix[0][1]*(y1) + Imatrix[0][2]*(z1));

hum->xRotCadera1 +=dtheta1;

dtheta2 = (Imatrix[1][0]*(x1) + Imatrix[1][1]*(y1) + Imatrix[1][2]*(z1));

hum->yRotCadera1 +=dtheta2;

```

```
dtheta3 = (Imatrix[2][0]*(x1) + Imatrix[2][1]*(y1) + Imatrix[2][2]*(z1));
```

```
hum->xRotRodilla1 +=dtheta3;
```

```
}
```

```
// Cinemática Inversa para la Pierna Derecha.
```

```
void AnimaPieDerecho(RtFloat x, RtFloat y, RtFloat z, humano hum)
```

```
{
```

```
    RtFloat x0, y0, z0, x1, y1, z1, det;
```

```
    RtFloat theta1, theta2, theta3, dtheta1, dtheta2, dtheta3;
```

```
    RtFloat lPierna, lColumna, wPelvis, wEspalda, lColumna1, lColumna2;
```

```
    RtFloat matrix[3][3], Imatrix[3][3]; //Jacobiano y su Inversa.
```

```
    lPierna = hum->lPierna/2;
```

```
    lColumna1 = hum->lColumna/2;
```

```
    lColumna2 = hum->lColumna/2 + 5;
```

```
    lColumna = hum->lColumna + 5;
```

```
    wEspalda = hum->wEspalda;
```

```
    wPelvis = hum->wPelvis;
```

```
    theta1 = hum->xRotCadera2/57.295f;
```

```
    theta2 = hum->yRotCadera2/57.295f;
```

```
    theta3 = hum->xRotRodilla2/57.295f;
```

```
        // Encuentra la Posición desde los ángulos actuales;
```

```
        // el pivote del tronco es (0, -30, 0)
```

```
    x0 = - wPelvis/2 - (lPierna*(float)sin(theta3 + theta1) +  
        lPierna*(float)sin(theta1))*(float)sin(theta2);
```

```
    y0 = -30 - (lPierna*(float)cos(theta3 + theta1) + lPierna*(float)cos(theta1));
```

```
    z0 = (lPierna*(float)sin(theta3 + theta1) +  
        lPierna*(float)sin(theta1))*(float)cos(theta2);
```

```

// forma el jacobiano [J] =
// [JaJb Jc] <- [d(x0)/d(theta1) d(x0)/d(theta2) d(x0)/d(theta3)]
// [JdJd Jf]   [d(y0)/d(theta1) d(y0)/d(theta2) d(y0)/d(theta3)]
// [JgJh Ji]   [d(z0)/d(theta1) d(z0)/d(theta2) d(z0)/d(theta3)]

matrix[0][0] = -(lPierna*(float)cos(theta3+theta1) +
                lPierna*(float)cos(theta1))*(float)sin(theta2);

matrix[0][1] = -(lPierna*(float)sin(theta3+theta1) +
                lPierna*(float)sin(theta1))*(float)cos(theta2);

matrix[0][2] = -lPierna*(float)cos(theta3+theta1)*(float)sin(theta2);

matrix[1][0] = -(-lPierna*(float)sin(theta3+theta1) - lPierna*(float)sin(theta1));

matrix[1][1] = 0;

matrix[1][2] = lPierna*(float)sin(theta3+theta1);

matrix[2][0] = (lPierna*(float)cos(theta3+theta1) +
                lPierna*(float)cos(theta1))*(float)cos(theta2);

matrix[2][1] = -(lPierna*(float)sin(theta3+theta1) +
                lPierna*(float)sin(theta1))*(float)sin(theta2);

matrix[2][2] = (lPierna*(float)cos(theta3+theta1))*(float)cos(theta2);

//                                     [deltax/n]
// Encuentra una Delta Pequeña r = [deltay/n]
//                                     [deltaz/n]

x1 = (x-x0)/0.1f;
y1 = (y-y0)/0.1f;
z1 = (z-z0)/0.1f;

// Resuelve la Inversa de la Matriz
// Usando la Regla de Cramer:
// J-1 = adj(J)/det(J)

// Calcula el Determinante de [J]
// Sí det = 0, no Existe Inversa.

```

```

det = (
    + matrix[0][0]*(matrix[1][1]*matrix[2][2] - matrix[2][1]*matrix[1][2])
    - matrix[0][1]*(matrix[1][0]*matrix[2][2] - matrix[2][0]*matrix[1][2])
    + matrix[0][2]*(matrix[1][0]*matrix[2][1] - matrix[2][0]*matrix[1][1])
);

```

// No Puede Dividir entre 0; si det == 0, no Existe Inversa -> no Cambian los ángulos.

```

if (det == 0) det = 1.0f;

```

```

// Inversa de [J] es (1/det)*adj(J)

```

```

Imatrix[0][0] = (matrix[1][1]*matrix[2][2] - matrix[1][2]*matrix[2][1])/det;
Imatrix[0][1] = -(matrix[0][1]*matrix[2][2] - matrix[0][2]*matrix[2][1])/det;
Imatrix[0][2] = (matrix[0][1]*matrix[1][2] - matrix[0][2]*matrix[1][1])/det;

```

```

Imatrix[1][0] = -(matrix[1][0]*matrix[2][2] - matrix[1][2]*matrix[2][0])/det;
Imatrix[1][1] = (matrix[0][0]*matrix[2][2] - matrix[0][2]*matrix[2][0])/det;
Imatrix[1][2] = -(matrix[0][0]*matrix[1][2] - matrix[0][2]*matrix[1][0])/det;

```

```

Imatrix[2][0] = (matrix[1][0]*matrix[2][1] - matrix[1][1]*matrix[2][0])/det;
Imatrix[2][1] = -(matrix[0][0]*matrix[2][1] - matrix[0][1]*matrix[2][0])/det;
Imatrix[2][2] = (matrix[0][0]*matrix[1][1] - matrix[0][1]*matrix[1][0])/det;

```

```

// Calcula dtheta = J-1 * r

```

```

//   [   ] [deltax/n] [dtheta1]
//   [ J-1 ] [deltay/n] = [dtheta2]
//   [   ] [deltaz/n] [dtheta3]

```

```

// Cambio en los ángulos.

```

```

// Entonces nueva_theta = theta + dtheta

```

```

dtheta1 = (Imatrix[0][0]*(x1) + Imatrix[0][1]*(y1) + Imatrix[0][2]*(z1));

```

```

hum->xRotCadera2 +=dtheta1;

```

```

dtheta2 = (Imatrix[1][0]*(x1) + Imatrix[1][1]*(y1) + Imatrix[1][2]*(z1));

```

```

hum->yRotCadera2 +=dtheta2;

dtheta3 = (Imatrix[2][0]*(x1) + Imatrix[2][1]*(y1) + Imatrix[2][2]*(z1));

hum->xRotRodilla2 +=dtheta3;

}

// Cinemática Inversa para el Brazo Izquierdo.
void AnimaArmIzquierdo(RtFloat x, RtFloat y, RtFloat z, humano hum)
{
    RtFloat x0, y0, z0, x1, y1, z1, det;
    RtFloat theta1, theta2, theta3, dtheta1, dtheta2, dtheta3;
    RtFloat lArm, lColumna, wEspalda, lColumna1, lColumna2;
    RtFloat matrix[3][3], Imatrix[3][3]; //Jacobiano y su Inversa.

    lArm = hum->lBrazo/2;
    lColumna1 = hum->lColumna/2;
    lColumna2 = hum->lColumna/2 + 5;
    lColumna = hum->lColumna + 5;
    wEspalda = hum->wEspalda;

    theta1 = hum->xRotBrazo1/57.295f;
    theta2 = hum->yRotBrazo1/57.295f;
    theta3 = hum->xRotCodo1/57.295f;

    x0 = - wEspalda/2 - 2*lArm*(float)sin(theta1) *(float)sin(theta2);

    y0 = -30 + lColumna1*(float)cos(theta3) + lColumna2*(float)cos(theta3) -
        (2*lArm*(float)cos(theta1-theta3));

    z0 = lColumna1*(float)sin(theta3)+lColumna2*(float)sin(theta3)
        +(2*lArm*(float)sin(theta1-theta3))*(float)cos(theta2);

    // forma el jacobiano [J] =
    // [JaJb Jc] [d(x0)/d(theta1) d(x0)/d(theta2) d(x0)/d(theta3)]
    // [JdJd Jf] [d(y0)/d(theta1) d(y0)/d(theta2) d(y0)/d(theta3)]
    // [JgJh Ji] [d(z0)/d(theta1) d(z0)/d(theta2) d(z0)/d(theta3)]

```

```

matrix[0][0] = -2*lArm*(float)cos(theta1) *(float)sin(theta2);
matrix[0][1] = -2*lArm*(float)sin(theta1) *(float)cos(theta2);
matrix[0][2] = 0;

matrix[1][0] = (2*lArm*(float)sin(theta1-theta3));
matrix[1][1] = 0;
matrix[1][2] = -lColumna1*(float)sin(theta3)-lColumna2*(float)sin(theta3)-
2*lArm*(float)sin(theta1-theta3);

matrix[2][0] = 2*lArm*(float)cos(theta1-theta3)*(float)cos(theta2);
matrix[2][1] = -2*lArm*(float)sin(theta1-theta3) *(float)sin(theta2);
matrix[2][2] = lColumna1*(float)cos(theta3)+lColumna2*(float)cos(theta3)-
(2*lArm*(float)cos(theta1-theta3))*(float)cos(theta2);

//                               [deltax/n]
//Encuentra una Delta Pequeña r = [deltay/n]
//                               [deltaz/n]

x1 = (x-x0)/0.1f;
y1 = (y-y0)/0.1f;
z1 = (z-z0)/0.1f;

// Resuelve la Inversa de la Matriz
// Usando la Regla de Cramer:
//  $J^{-1} = \text{adj}(J)/\det(J)$ 
// Calcula el Determinante de [J]
// Sí det = 0, no Existe Inversa.

det = (
    + matrix[0][0]*(matrix[1][1]*matrix[2][2] - matrix[2][1]*matrix[1][2])
    - matrix[0][1]*(matrix[1][0]*matrix[2][2] - matrix[2][0]*matrix[1][2])
    + matrix[0][2]*(matrix[1][0]*matrix[2][1] - matrix[2][0]*matrix[1][1])
);

// No Puede Dividir entre 0; Sí det == 0, no Existe Inversa -> no Cambian los ángulos.

if (det == 0) det = 1.0f;

// inversa de [J] es (1/det)*adj(J)

```

```

Imatrix[0][0]=(matrix[1][1]*matrix[2][2] - matrix[1][2]*matrix[2][1])/det;
Imatrix[0][1]=-(matrix[0][1]*matrix[2][2] - matrix[0][2]*matrix[2][1])/det;
Imatrix[0][2]=(matrix[0][1]*matrix[1][2] - matrix[0][2]*matrix[1][1])/det;

```

```

Imatrix[1][0]=-(matrix[1][0]*matrix[2][2] - matrix[1][2]*matrix[2][0])/det;
Imatrix[1][1]=(matrix[0][0]*matrix[2][2] - matrix[0][2]*matrix[2][0])/det;
Imatrix[1][2]=-(matrix[0][0]*matrix[1][2] - matrix[0][2]*matrix[1][0])/det;

```

```

Imatrix[2][0]=(matrix[1][0]*matrix[2][1] - matrix[1][1]*matrix[2][0])/det;
Imatrix[2][1]=-(matrix[0][0]*matrix[2][1] - matrix[0][1]*matrix[2][0])/det;
Imatrix[2][2]=(matrix[0][0]*matrix[1][1] - matrix[0][1]*matrix[1][0])/det;

```

```

// Calcula dtheta = J-1 * r

```

```

// [   ] [deltax/n] [dtheta1],
// [ J^-1 ] [deltay/n] = [dtheta2]
// [   ] [deltaz/n] [dtheta3]

```

```

// Cambio en los ángulos.
// Entonces nueva_theta = theta + dtheta

```

```

dtheta1 = (Imatrix[0][0]*(x1) + Imatrix[0][1]*(y1) + Imatrix[0][2]*(z1));

```

```

hum->xRotBrazo1 +=dtheta1;

```

```

dtheta2 = (Imatrix[1][0]*(x1) + Imatrix[1][1]*(y1) + Imatrix[1][2]*(z1));

```

```

hum->yRotBrazo1 +=dtheta2;

```

```

dtheta3 = (Imatrix[2][0]*(x1) + Imatrix[2][1]*(y1) + Imatrix[2][2]*(z1));

```

```

hum->xRotCodo1 +=dtheta3;

```

```

}

```

```

// Cinemática Inversa para el Brazo Derecho.

```

```

void AnimaArmDerecho(RtFloat x, RtFloat y, RtFloat z, humano hum)

```

```

{

```

```

RtFloat x0, y0, z0, x1, y1, z1, det;

```

```

RtFloat theta1, theta2, theta3, dtheta1, dtheta2, dtheta3;

```

```

RtFloat lArm, lColumna, wEspalda, lColumna1, lColumna2;

```

```
RtFloat matrix[3][3], Imatrix[3][3]; //Jacobiano y su Inversa.
```

```
lArm = hum->lBrazo/2;
lColumna1 = hum->lColumna/2;
lColumna2 = hum->lColumna/2 + 5;
lColumna = hum->lColumna + 5;
wEspalda = hum->wEspalda;
```

```
theta1 = hum->xRotBrazo2/57.295f;
theta2 = hum->yRotBrazo2/57.295f;
theta3 = hum->xRotCodo2/57.295f;
```

```
// Encuentra la Posición desde los ángulos actuales;
// el pivote del tronco es (0, -30, 0)
```

```
x0 = - wEspalda/2 - 2*lArm*(float)sin(theta1) *(float)sin(theta2);
```

```
y0 = -30 + lColumna1*(float)cos(theta3) + lColumna2*(float)cos(theta3) -
(2*lArm*(float)cos(theta1-theta3));
```

```
z0 = lColumna1*(float)sin(theta3)+lColumna2*(float)sin(theta3)+
(2*lArm*(float)sin(theta1-theta3))*(float)cos(theta2);
```

```
// forma el jacobiano [J] =
```

```
// [JaJb Jc] <- [d(x0)/d(theta1) d(x0)/d(theta2) d(x0)/d(theta3)]
// [JdJd Jf] [d(y0)/d(theta1) d(y0)/d(theta2) d(y0)/d(theta3)]
// [JgJh Ji] [d(z0)/d(theta1) d(z0)/d(theta2) d(z0)/d(theta3)]
```

```
matrix[0][0] = -2*lArm*(float)cos(theta1) *(float)sin(theta2);
matrix[0][1] = -2*lArm*(float)sin(theta1) *(float)cos(theta2);
matrix[0][2] = 0;
```

```
matrix[1][0] = (2*lArm*(float)sin(theta1-theta3));
matrix[1][1] = 0;
matrix[1][2] = -lColumna1*(float)sin(theta3)-lColumna2*(float)sin(theta3)-
2*lArm*(float)sin(theta1-theta3);
```

```
matrix[2][0] = 2*lArm*(float)cos(theta1-theta3)*(float)cos(theta2);
matrix[2][1] = -2*lArm*(float)sin(theta1-theta3) *(float)sin(theta2);
matrix[2][2] = lColumna1*(float)cos(theta3)+lColumna2*(float)cos(theta3)-
```

```

(2*IArm*(float)cos(theta1-theta3))*(float)cos(theta2);

//                               [deltax/n]
//Encuentra una Delta Pequeña r = [deltay/n]
//                               [deltaz/n]

x1 = (x-x0)/0.1f;
y1 = (y-y0)/0.1f;
z1 = (z-z0)/0.1f;

// Resuelve la Inversa de la Matriz
// Usando la Regla de Cramer:
//  $J^{-1} = \text{adj}(J)/\det(J)$ 
// Calcula el Determinante de [J]
// Si det = 0, no Existe Inversa.

det = (
    + matrix[0][0]*(matrix[1][1]*matrix[2][2] - matrix[2][1]*matrix[1][2])
    - matrix[0][1]*(matrix[1][0]*matrix[2][2] - matrix[2][0]*matrix[1][2])
    + matrix[0][2]*(matrix[1][0]*matrix[2][1] - matrix[2][0]*matrix[1][1])
);

// No Puede Dividir entre 0; Si det == 0, no Existe Inversa -> no Cambian los ángulos.

if (det == 0) det = 1.0f;

// Inversa de [J] es (1/det)*adj(J)

Imatrix[0][0]=(matrix[1][1]*matrix[2][2] - matrix[1][2]*matrix[2][1])/det;
Imatrix[0][1]=- (matrix[0][1]*matrix[2][2] - matrix[0][2]*matrix[2][1])/det;
Imatrix[0][2]=(matrix[0][1]*matrix[1][2] - matrix[0][2]*matrix[1][1])/det;
Imatrix[1][0]=- (matrix[1][0]*matrix[2][2] - matrix[1][2]*matrix[2][0])/det;
Imatrix[1][1]=(matrix[0][0]*matrix[2][2] - matrix[0][2]*matrix[2][0])/det;
Imatrix[1][2]=- (matrix[0][0]*matrix[1][2] - matrix[0][2]*matrix[1][0])/det;

Imatrix[2][0]=(matrix[1][0]*matrix[2][1] - matrix[1][1]*matrix[2][0])/det;
Imatrix[2][1]=- (matrix[0][0]*matrix[2][1] - matrix[0][1]*matrix[2][0])/det;
Imatrix[2][2]=(matrix[0][0]*matrix[1][1] - matrix[0][1]*matrix[1][0])/det;

// Calcula dtheta = J-1 * r
//   [      ][deltax/n] [dtheta1],
//   [ J-1 ][deltay/n] = [dtheta2]
//   [      ][deltaz/n] [dtheta3]

```

```

        // Cambio en los ángulos.
        // Entonces nueva_theta = theta + dtheta

        dtheta1 = (Imatrix[0][0]*(x1) + Imatrix[0][1]*(y1) + Imatrix[0][2]*(z1));

        hum->xRotBrazo2 +=dtheta1;

        dtheta2 = (Imatrix[1][0]*(x1) + Imatrix[1][1]*(y1) + Imatrix[1][2]*(z1));

        hum->yRotBrazo2 +=dtheta2;

        dtheta3 = (Imatrix[2][0]*(x1) + Imatrix[2][1]*(y1) + Imatrix[2][2]*(z1));

        hum->xRotCodo2 +=dtheta3;

    }

```

// Cinemática Directa para el Movimiento de la Cadera,
// la Pierna Posterior y el Avance del Esqueleto.

```

void MovimientoCadera(humano hum, bool izquierdo) {
    if (izquierdo == true) {
        hum->xRotBrazo2 -=1.0;
        hum->xRotCadera2 -= 1.59999;
        hum->Avance+= 1.9;
        hum->RotPelvis += 1.6;
    }
    else {
        hum->xRotBrazo1 -=1.0;
        hum->xRotCadera1 -=1.59999;
        hum->Avance+= 1.09;
        hum->RotPelvis -= 1.6;
    }
}

```

// Programa principal.

```

int main(int argC, char** argV)
{

```

```

    char nombearchivo[20];

```

ESTA TESIS NO SALE
DE LA BIBLIOTECA

```

char* renderer = RI_NULL;

humano h;
RtFloat a1,b1,c1, a2,b2,c2;
RtFloat d1,e1,f1, d2,e2,f2;

RtString encendido = "on\0";

int cuadro = 0, j;

if (argC != 2) {
#if defined(_WIN32)
    fprintf(stderr,
            "Escriba: %s RIB|rgl|rendrib\n\n",
            argV[0]);
#else
    fprintf(stderr,
            "Escriba: %s RIB|rgl|rendrib|rendribv\n\n",
            argV[0]);
#endif
    exit (-1);
}

// Coordinadas para las Fuentes de Luz
renderer = argV[1];
RiBegin(renderer);
RtPoint p1= {-1,0,0};
RtFloat kd =0.75;
RtFloat kr=0.25;
RtFloat km=0.5;
RtFloat a = 0.0f;
h = humanoInit(a);

// Ciclo de la Secuencia de Animación.
for (int i = 1; i < 9; i++) {

    if (i % 2) {

        // PI Coordinada a donde tiene que llegar la Pierna Izquierda.
        a1 = -10;
        b1 = -86;
        c1 = 21;
    }
}

```

```

// BI Coordenada a donde tiene que llegar el Brazo Izquierdo
    d1 = -20;
    e1 = -19;
    f1 = -14;

// PD Coordenada a donde tiene que llegar la Pierna Derecha
    a2= -10;
    b2 = -86;
    c2 = -21;

// BD Coordenada a donde tiene que llegar el Brazo Derecho
    d2 = -20;
    e2 = -21;
    f2 = 11;
        j = 1;
    }
        else { // Cambia Puntos de llegada en Piernas y Brazos

            f1 = 11;
            f2 = -14;
            c1 = -21;
            c2 = 21;
            j = 0;
        }

// Empezamos con la Pierna y Brazo Izquierdo la Animación

for (cuadro = 0; cuadro <= 30; cuadro++) {
    sprintf(nombrearchivo, "kn#%d.tif", (cuadro + (i-1)+(i-1)*30 + 100));
        RiFrameBegin(cuadro);
            RiDisplay(nombrearchivo, RI_FILE, RI_RGB, RI_NULL);
            RiProjection("perspective", RI_NULL);
            RiTranslate(-60.0, 30, 300.0);
            RiRotate(90.0, 0.0, 1.0, 0.0);

            RtColor color;
            color[0] = 1.0f;
            color[1] = 1.0f;
            color[2] = 1.0f;
            RiColor(color);

```

```

RiWorldBegin();
    RiLightSource("ambientlight", RI_NULL);
    RiAttribute("light", "shadows", &encendido, RI_NULL);
    RiLightSource("distantlight", "to", &p1, RI_NULL);
    RiDeclare("kd", "uniform float");
    RiAttributeBegin();
    RiSurface("plastic", RI_NULL);
    color[0] = 1.0f;
        color[1] = 1.0f;
        color[2] = 1.0f;
        RiColor(color);
        RiPolygon(4, RI_P, (RtPointer) plado, RI_NULL);
    RiAttributeEnd();
    RiAttributeBegin();
        RiSurface("matte", "kd", (RtPointer) &kd, RI_NULL);
        color[0] = 1.0f;
        color[1] = 1.0f;
        color[2] = 1.0f;
        RiColor(color);

        RiAttributeBegin();
            color[0] = 0.64f;
            color[1] = 0.15f;
            color[2] = 0.1f;
            RiColor(color);
            RiSurface("brick", RI_NULL);
            RiTextureCoordinates(0,0,1.7,0,0,1.7,1.7,1.7);
            RiPatch("bilinear", RI_P, (RtPointer) pfte, RI_NULL
            );
        RiAttributeEnd();

        RiAttributeBegin();
            RiDeclare("kr", "uniform float");
            RiDeclare("km", "uniform float");
            RiSurface("superplank", "kr", (RtPointer) &kr, "km
            ", (RtPointer) &km, RI_NULL);
            color[0] = 0.5;
            color[1] = 0.2f;
            color[2] = 0.67f;
            RiColor(color);
            RiPatch("bilinear", RI_P, (RtPointer) piso,
            RI_NULL);
        RiAttributeEnd();
    RiAttributeEnd();

Restricciones(h);

```

```

ModeloHumano(h);
if (!(cuadro % 2)) {

    MovimientoCadera(h, true);
    AnimaArmIzquierdo(d1, e1, f1, h);

    if (j == 0) // rojo
    {
        Restricciones1(h);
        AnimaPieDerecho(a2, b2, c2, h);
// Cinemática Inversa de los Pies Izquierdo, derecho
        h->xRotTobillo1 = h->xRotCadera1 + h->xRotRodilla1;
        h->xRotTobillo2 = 0;} // fin (j ==0)
    } // cuadro % 2
    else {

        MovimientoCadera(h, false);
        AnimaArmDerecho(d2, e2, f2, h);
        if (j == 1) // azul
        { // Apoyo en Inicio
            Restricciones2(h);
            AnimaPieIzquierdo(a1, b1, c1, h);
// Cinemática Inversa de los Pies Derecho, izquierdo
            h->xRotTobillo2 = h->xRotCadera2 + h->xRotRodilla2;
            h->xRotTobillo1 = 0;} // fin if (j ==1)

        } // fin if (!(cuadro % 2))

        RiWorldEnd();

        RiFrameEnd();

    } // fin for (cuadro) poses de animación

} // fin for (i) secuencia de animación

RiEnd();
return 1;
}

```

Bibliografía.

[AG85] W Armstrong and M Green The dynamics of articulated rigid bodies for purposes of animation The Visual Computer, 1(4):231-240, 1985

[Bad75] Norman I Badler Temporal scene analysis: Conceptual description of object movements. PhD thesis, Computer Science, Univ. of Toronto, Toronto, Canada, 1975 (Univ. of Pennsylvania, Computer and Information Science, Tech Report MS-CIS-76-4)

[Bad99] Norman I. Badler, Cary B. Phillips, Bonnie L. Webber. Simulating Humans: Computer graphics, animation and control. Oxford Univ Press, March 25, 1999. (Univ. of Pennsylvania Philadelphia, Computer and Information Science)

[Bai86] J Bailleul. Avoiding obstacles and resolving kinematic redundancy In Proc 1986 Int. Conf. on Robotics and Automation, April 1986

[BB87] R Barzel and A Barr Modeling with dynamic constraints In SIGGRAPH Course Notes: Topics in Physically - Based Modelling, 1987

[BC89] A Bruderlin and T Calvert Goal-directed dynamic animation of human walking Computer Graphics, 23(3):233-242, July 1989

[BKK+85] Norman I. Badler, Jonathan Korein, James U. Korein, Gerald Radack, and Lynne Brotman. Positioning and animating human figures in a task-oriented environment. The Visual Computer, 1(4):212-220, 1985.

[Bli82] James F. Blinn. A generalization of algebraic surface drawing ACM Transactions on Graphics, 1(3):235-256, July 1982

[BMB86] Norman I. Badler, Kamran H. Manoochehri, and David Baraff Multi-dimensional input techniques and articulated figure positioning by multiple constraints. In Proc. Workshop on Interactive 3D Graphics, New York, NY, Oct 1986 ACM

[BMW87] Norman I. Badler, Kamran Manoochehri, and G. Walters Articulated figure positioning by multiple constraints IEEE Computer Graphics and Applications, 7(6):28-38, June 1987

- [BN88] L. Brotman and A. Netravali. Motion interpolation by optimal control. *Computer Graphics*, 22(4):309-315, August 1988
- [BOK80] Norman I. Badler, Joseph O'Rourke, and Bruce Kaufman. Special problems in human movement simulation. *Computer Graphics*, 14(3):189-197, July 1980
- [BOI79] Norman I. Badler, Joseph O'Rourke, and Hasida Ioltzis. A spherical representation of a human body for visualizing movement. *IEEE Proceedings*, 67(10):1397-1403, Oct. 1979.
- [Bru88] Armin Bruderlin. Goal-directed, dynamic animation of bipedal locomotion. Master's thesis, Simon Fraser Univ., Vancouver, Canada, 1988
- [BS79] Norman I. Badler and Stephen W. Smoliar. Digital representations of human movement. *ACM Computing Surveys*, 11(1):19-38, March 1979
- [BSOW78] Norman I. Badler, Stephen W. Smoliar, Joseph O'Rourke, and Lynne Webber. The simulation of human movement. Technical Report MS-CIS-78-36, Computer and Information Science, Univ of Pennsylvania, Philadelphia, PA, 1978
- [BwDL80] Irmgard Bartenieff and (with Dori Lewis). *Body movement: Coping with the environment*. Gordon and Breach, New York, NY, 1980
- [BWKE91] Norman I. Badler, Bonnie L. Webber, Jugal K. Kalita, and Jeffrey Esakov. Animation from instructions. In Norman I. Badler, Brian A. Barsky, and David Zeltzer, editors, *Making them move: Mechanics, control, and animation of articulated figures*, pages 51-93. Morgan-Kaufmann, San Mateo, CA, 1991
- [Cal91] Tom Calvert. Composition of realistic animation sequences for multiple human figures. In Norman I. Badler, Brian A. Barsky, and David Seltzer, editors, *Making them move: Mechanics, control, and animation of articulated figures*, pages 35-50. Morgan-Kaufmann, San Mateo, CA, 1991
- [Cat78] E. Catmull. The problems of computer-assisted animation. *Computer Graphics*, 12(3):348-353, August 1978
- [FW88] D. Forsey and J. Wilhelms. Techniques for interactive manipulation of articulated bodies using dynamic analysis. In *Proceedings of Graphics Interface*, pages 8-15, 1988
- [Gir91] M. Girard. Constrained optimization of articulated animal movement in computer animation. In Norman I. Badler, Brian A. Barsky, and David Zeltzer, editors, *Making Them Move: Mechanics, Control, and Animation of Articulated Figures*, chapter 10, pages 209-229. Morgan Kaufmann Publishers, Inc., San Mateo, Ca., 1991.

- [Gir87] Michael Girard Interactive design of 3D computer-animated legged animal motion IEEE Computer Graphics and Applications, 7(6):39-51, 1987
- [GM96a] George Maestri, [Digital] Character animation, chapter 2 "Modeling Bodies for Animation", page 28-31, News Riders Publishing, Indianapolis Indiana, 1996
- [GM96b] George Maestri, [Digital] Character animation, chapter 7 "Walking and Locomotion", page 205-237, News Riders Publishing, Indianapolis Indiana, 1996
- [Hah88] James K Hahn Realistic animation of rigid bodies. Computer Graphics, 22(4):299-308, August 1988
- [Hog84] DC Hogg, GD Sullivan, KD Baker and DH Mott Recognition of vehicles in traffic using geometric models IEE International Conference on Road Traffic Data Collection, London, IEEE 1984
- [IC88] P. Isaacs and M. Cohen Mixed methods for complex kinematic constraints in dynamic figure animation The Visual Computer, 4:296-305, 1988
- [Igo99] Tolga K, Igor S., Nadia T, Daniel T Avatars in Networked Virtual Environments: John Wiley & Sons, LTD 1999
- [Kak97] I Kakadiaris, D. Metaxas, R. Bajcsy. Inferring 2D Object Structure from the Deformation of Apparent Contours Computer Vision and Image Understanding, 65(2):129-147, February 1997
- [Mac90] A. A. Maciejewski. Dealing with the ill-conditioned equations of motion for articulated figures IEEE Computer Graphics and Applications, 10(3):233-242, May 1990
- [Mar85] David Marr Vision : a Computational investigation into the human representation and processing of visual information Madrid: Alianza, 1985
- [MA92] Mark Watt Alan Watt. Advanced animation and rendering techniques theory and practice, Addison Wesley, 1992
- [MK95] Myoung-Jun Kim and Myung-Soo Kim and Sung Yong Shin A General Construction Scheme for Unit Quaternion Curves with Simple High Order Derivatives In Robert Cook, editor, Computer Graphics (SIGGRAPH '95 Proceedings), volume 29, pages 369-376, August 1995
- [MTT88a] Nadia Magnenat-Thalmann and Daniel Thalmann Introduction to biped walking, SIGGRAPH88, 1988.

- [MTT88b] Nadia Magnenat-Thalmann and Daniel Thalmann Mechanics and robotics for animation synthetic actors, SIGGRAPH88, 1988
- [MTT90] Nadia Magnenat-Thalmann and Daniel Thalmann Synthetic actors in 3-D computer-generated films. Springer-Verlag, New York, NY, 1990
- [Roh94] K. Rohr Towards Model-Based Recognition of Human Movements in Image Sequences CVGIP, Image Understanding, 59(1):94, 1994
- [RP2002] Rick Parent. Computer animation algorithms and techniques Morgan-Kaufmann, Berkeley, CA, 2002
- [SM2001] Harold C Sun, Dimitris N Metaxas, Automating gait generation, University of Pennsylvania, ACM SIGGRAPH 2001, 12-17 August 2001
- [SJ99] Shih Kai Chung and James K Hahn, Animation of human walking in virtual environments, George Washington University, IEEE Proceedings of the Computer Animation, 1999
- [Tcb89] Thomas W Calvert Armin Bruderlin, Goal – directed dynamic animation of human walking, Computer Graphics, 23(3): 233-242, July 1989
- [WCH88] B Wyvill, M Chmilar and C Herr A simple model of human animation. In SIGGRAPH Course Notes: Synthetic actors: The Impact of Artificial Intelligence and Robotics on Animation, 1988
- [Wil86] J Wilhelms. Virya – a motion control editor for kinematic and dynamic animation In Proceedings of Graphics Interface, pages 141-146, 1986
- [Wil87] Jane Wilhelms Using dynamic analysis for realistic animation of articulated bodies IEEE Computer Graphics and Applications, 7(6):12-27, 1987
- [Wil91] J Wilhelms Dynamic experiences. In N Badler, B Barsky, and D Zeltzer, editors, Making them move: Mechanics, control, and animation of articulated figures, chapter 13, pages 265-280 Morgan Kaufmann Publishers, Inc., San Mateo, Ca, 1991.
- [Win90] David A Winter Biomechanics and motor control of human movement Wiley Interscience, New York, NY, 1990 Second Edition
- [WK88] A. Witkin and M. Kass. Spacetime constraints Computer Graphics, 22(4):159-168, August 1988.
- [Zel82] D. Zeltzer Motor control techniques for figure animation IEEE Computer Graphics and Applications, 2(9):53-59, 1982

[Zel91] David Zeltzer. Task-level graphical simulation: Abstraction, representation, and control. In Norman I Badler, Brian A Barsky, and David Seltzer, editors, *Making them move: Mechanics, control, and animation of articulated figures*, pages 3-33. Morgan-Kaufmann, San Mateo, CA, 1991.

