

UNIVERSIDAD POLITÉCNICA DE VALENCIA

DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN



INTERACCIÓN MULTITÁCTIL EN UN SISTEMA DE GENERACIÓN PROCEDURAL

Master en

Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital

Armando de la Re Vega

Dirigido por

Emilio Camahort Gurrea

Francisco José Abad Cerdá

Julio 2009

Agradecimientos

Esta tesis la dedico a todos aquellos que me brindaron su cooperación desinteresada, sin las cuales no hubiese sido posible su finalización. Muchos han sido un soporte muy fuerte en momentos de angustia y desesperación.

Le doy gracias a Dios por estar conmigo en cada momento, por fortalecer mi corazón, por iluminar mi mente y darme fuerzas para seguir adelante. Por haber puesto en mi camino a todas aquellas personas que me han servido de soporte y compañía. Sobre todo por permitirme continuar estudiando y superarme.

Agradezco al CONACYT por brindarme de los recursos monetarios para estudiar este master, sobre todo por exigirme altos requisitos y por cumplir con el apoyo que prometió al asignarme la beca de estudios.

Mi más sincero agradecimiento a Emilio Camahort por sus consejos, guía y ayuda al ser mi director de tesis. A Paco Abad por su paciencia, si sobre todo por ella. Por soportar mis apuros, mala ortografía y gramática, problemas de programación y construcción del hardware, ausencia de conocimientos, a ayudarme a pensar e iluminarme cuando veía todo perdido. A Eduardo por ser mi compañero de construcción y su ayuda.

Agradezco a mi familia, porque a pesar de no estar presentes físicamente y de encontrarse al muy lejos, en mi país México, han estado presentes en mente y corazón. Está claro que si no fuese por el esfuerzo realizado por ellos, sus consejos, apoyo incondicional y constante preocupación, este trabajo no hubiese sido posible. A mis padres Enrique y Bertha María, a los que acudo cuando me encuentro en esos momentos difíciles. Los primeros en auxiliarme, los primeros en cargar mis problemas. Mi más grande amor a los dos. A mi manina Bertha, que ha pasado por algunos momentos difíciles y me hubiera gustado estar ahí con ella. Siempre se preocupa por mí, y la quiero muchísimo. Mis tías, Lolita, Guillermina, Panchi y Elodia que sé siempre estoy en sus oraciones. A mí tío Jesús, quien ya no está con nosotros, pero sí en mi corazón. Sé que me apoyaste mucho y siempre creíste en mi para lograr mi sueño, desde la universidad hasta el último momento. Te dedico este gran logro. A la Carmelita... mí Carmelita. Tantos correos, mensajes, llamadas, SMS, cajas, cartas, paquetes. Siempre tan pendiente de mí, por tu apoyo incondicional, tus consejos, confianza, tu amor, preocupaciones, tus hombros donde llorar y tus lagrimas. Mi madrina, mi otra madre, mi amiga. Por compartir todo conmigo. A mis hermanos Enrique, Guillermo y Sandra, quienes siempre han sido mis cómplices, confidentes, apoyos y ejemplos a seguir, pero sobre todo por ayudarme cuando más lo necesito. A todos por compartir mis logros y alegrías, por ayudarme a salir de mis penas y problemas. No saben cómo les extraño.

A mis familiares postizos, quienes no sé como soportan tanta depresión en mis momentos más oscuros. A Sonia por sus consejos desinteresados, sus regaños y su paciencia a recibirme siempre y ayudarme a salir adelante; a Paula que siempre me da ánimos por más mal que se encuentre, por preocuparse por mi vida y buscarme alternativas a mis baches; a las dos por los momentos de risas al por mayor. Mis "guías" en España. A Andrés por ayudarme a conseguir lo necesario en mis trabajos, soportarme en su casa cuando quisiera estar descansado, por ser mi cómplice de películas malas en el cine; a Víctor, por animarme a continuar siempre en el

camino del estudio, por brindarme su apoyo aún cuando no me conocía bien. A los cuatro, por esa amistad tan grande y hermosa que me brindaron desde el comienzo, gracias a la cual no dejan que me sienta solo, por lo cual nunca podré pagárselos y siempre estaré agradecido.

A Gustavo y Francisco, quienes desde el principio soportaron todas mis dudas y quebradas de cabeza, por la gran cantidad de sesiones de estudio, de explicaciones, de trabajos y programas. Por soportar mi dificultad de aprendizaje en ciertas materias, el ayudarme a sacar todo adelante. Estoy seguro que sin su ayuda aún seguiría atorado en algunas materias. Por “ayudarme a sacar el carro del fango” como alguna vez dijimos. Y aunado a todo eso, me soportaban en sus tiempos libres, brindándome su amistad y gratos momentos de diversión. Por ser perfectos compañeros, grandes amigos y excelentes personas.

A todos los amigos que hice a lo largo de este tiempo, que han estado ahí para apoyarme en todo momento, aunque algunos ya no estén físicamente conmigo. Fernanda, Martha, Marlene, Axel, Roxane, Carlos, Aremy, muchísimas gracias. A mis amigos que siempre están pendientes de mí, conociéndome como soy. Luis, Alex, René, Erika, Rafa, Pablo, Cecy, Lupita, Carlos, Arón, Lucero, Adriana, Pedro, Fede, Lucy, Laura, Patricia.

En general quisiera agradecer a todas y cada una de las personas que han vivido conmigo la realización de estos estudios y tesis, con sus altos y bajos, y que no necesito de nombrar porque sabemos que desde lo más profundo de mi corazón les agradezco el estar ahí conmigo.

Índice

1 Introducción	8
1.1 Resumen.....	8
1.2 Sistemas Multitáctil.....	10
1.3 Generación Procedural.....	11
1.4 Estructura del trabajo.....	12
2 Estado del Arte	13
2.1 Interfaces Multitáctiles	13
2.2 Tecnologías Multitáctiles	13
2.2.1 Reflexión Interna Total Frustrada (FTIR).	14
2.2.2 Iluminación Difusa (DI).	15
2.2.3 Plano de Luz Laser (LLP).	16
2.2.4 Iluminación Difusa de Superficie (DSI).	17
2.2.5 Plano de Luz LED (LED-LP)	18
2.3 Trabajos previos	19
2.4 Generación Procedural.....	21
2.4.1 Fractales	21
2.4.2 Sistemas L.....	22
2.5 Ejemplos de generadores procedurales.....	23
3 Análisis.....	26
3.1 Descripción del problema	26

3.2 Justificación	27
3.3 Objetivos	27
3.4 Resultados y Limitaciones	28
4 Implementación de interfaces multitáctiles	29
4.1 Banco de trabajo multitáctil	29
4.2 Sistema multitáctil portátil.....	35
5 Implementación de un generador procedural multitáctil	41
5.1 Estudio Previo.....	41
5.2 Implementación de un sistema de generación procedural multitáctil	45
6 Resultados.....	52
6.1 Sistema Multitáctil	52
6.1.1 Banco de trabajo multitáctil.....	52
6.1.2 Sistema multitáctil portátil.....	53
6.2 Generación Procedural.....	53
6.2.1 Sistemas de generación procedural actuales.....	53
6.2.2 Sistema de generación procedural multitáctil	54
7 Conclusiones.....	55
8 Trabajo Futuro	56
9 Publicaciones.....	58
10 Bibliografía	59
11 Anexos.....	63
11.1 Glosario	63
11.2 Artículos	65

Índice de Figuras

Figura 1: Estructura de una pantalla con Reflexión Interna Total Frustrada.	14
Figura 2: Estructura de una pantalla con iluminación Difusa Trasera.	15
Figura 3: The Reactable.	16
Figura 4: Estructura de una pantalla con Plano de Luz Laser.	17
Figura 5: Estructura de una pantalla con Iluminación Difusa de Superficie.	17
Figura 6: Estructura de una pantalla con un Plano de Luz LED.	18
Figura 7: Sintetizador de Música.	19
Figura 8: The CubTile prototipe. Interfáce de manejo de aplicacions 3D.	19
Figura 9: Reconocimiento de infrarrojo y sombras, el dedo tocando y el dedo cerca de la superficie.	20
Figura 10: Una imagen Fractal.	21
Figura 11: Árboles generados con Sistemas L.	22
Figura 12: Textura a base de Funciones de Ruido.	23
Figura 13: Modelado procedural de ciudades.	24
Figura 14: Modelado de una ciudad en un terreno.	24
Figura 15: Izquierda: Fotografía de una planta. Derecha: Modelo de planta basado en la Imagen.	25
Figura 16: Izquierda. Lijado del perfil del acrílico. Derecha. Creación del marco de madera.	30
Figura 17: Armado del cableado, leds infrarrojos y resistencias.	30
Figura 18: Cableado y leds infrarrojos ubicados a los dos lados del acrílico.	31
Figura 19: Izquierda. Aplicación de la capa de silicón. Derecha. Primer prueba.	31
Figura 20: Retirada del filtro infrarrojo.	32
Figura 21: Filtro paso alto de 850nm montado.	33
Figura 22: Mesa multitáctil en funcionamiento.	33
Figura 23: Pruebas de sensibilidad en la mesa multitáctil.	34
Figura 24: Banco de trabajo multitáctil finalizado.	34
Figura 25: Diseño básico de la pantalla.	35
Figura 26: Monitor desarmado.	36
Figura 27: Iluminación trasera difusa.	38
Figura 28: Estructura de las placas y el LCD.	39
Figura 29: Cableado interno y problemas con extensión de cables.	39
Figura 30: Detalles de la pantalla multitáctil. Ventilador, VGA e Interruptores.	40
Figura 31: La pantalla multitáctil terminada y funcionando.	40
Figura 32: Una producción final de F+F-F y una F+[F]-[F] con un ángulo de 45 grados.	46
Figura 33: Dos tipos de sistemas L modelados ("Simple plant" y "Nice tree").	47
Figura 34: Los árboles son animados y tienen los puntos donde se puede agregar hojas.	49
Figura 35: Animación con interacción multitáctil para formar árboles.	50

1 Introducción

1.1 Resumen

Históricamente la interacción con un ordenador se ha basado en un monitor, un teclado y un ratón. Por otro lado, las pantallas táctiles tratan de combinar esos tres dispositivos en uno sólo, para facilitar el uso del sistema a usuarios no acostumbrados a interactuar con ordenadores. Este tipo de pantallas se usan normalmente en lugares públicos, en los que es probable puedan ser utilizados por todo tipo de usuarios. Una de las características que hace especialmente amigable este tipo de sistemas es que la interacción es directa: el usuario “toca” el objeto sobre el que quiere interactuar. Un ejemplo de interacción indirecta es el uso de un ratón, que se usa para mover un puntero sobre la pantalla para interactuar con las aplicaciones. Otra limitación, tanto del ratón tradicional como de los dispositivos táctiles es que permiten un único punto de contacto con el sistema en cada momento.

La tecnología multitáctil ha comenzado a introducirse en nuestra vida cotidiana, impulsada principalmente por su exitosa inclusión en los teléfonos móviles de última generación. Los dispositivos multitáctiles se caracterizan por permitir al usuario utilizar simultáneamente más de un punto de contacto para interactuar con el sistema. Así ofrecen una serie de operaciones de más alto nivel. Un ejemplo típico de interacción con un interfaz multitáctil es permitir al usuario rotar y hacer zoom sobre imágenes dispuestas en la pantalla.

La aceptación de la tecnología multitáctil en los dispositivos móviles ha provocado que algunas compañías hayan lanzado al mercado pantallas de ordenador multitáctiles, ofreciendo al usuario formas alternativas de interacción con el ordenador. Sin embargo, esta tecnología emergente tiene problemas como su alto coste y el tamaño

de los dispositivos. Dichos problemas la restringen habitualmente a ferias, empresas o laboratorios de investigación.

Recientemente, grupos de entusiastas e investigadores han publicado algunas técnicas, manuales y vídeos de cómo construir pantallas multitáctiles. El problema es que la mayoría están basadas en un proyector, por lo que normalmente ocupan demasiado espacio.

En este trabajo presentamos dos soluciones multitáctiles, orientadas a dos tipos de usuarios: usuarios profesionales y usuarios casuales. Para el usuario profesional, se presenta un banco de trabajo que ofrece una superficie de interacción de grandes dimensiones. Para el usuario casual, presentamos una solución de tamaño reducido, portátil y de bajo coste. Para demostrar la utilidad de estos sistemas hemos desarrollado un pequeño prototipo dentro de un área de la Informática Gráfica con gran auge: la generación procedural.

Las aplicaciones gráficas actuales necesitan por lo general escenas de bastante complejidad, sobre todo cuando se trata de reproducir escenas naturales. Las técnicas procedurales permiten generar automáticamente objetos complejos definiendo un conjunto pequeño de reglas y seleccionando ciertos parámetros. Este proceso permite agilizar el proceso de creación de contenido y crear objetos al vuelo cuando son necesarios. La generación de escenas bajo demanda ofrece por tanto la posibilidad de crear mundos infinitos, lo que permite crear juegos para un gran número de usuarios simultáneos, que pueden explorar sin límites un escenario que nunca se repetirá.

Una de las grandes dificultades de los generadores procedurales de vegetación disponibles en la actualidad es la complejidad de su manejo. En este trabajo se presentará un estudio de las características de interacción y usabilidad de los sistemas actuales de mayor uso que implementan técnicas de modelado de plantas y fenómenos naturales con generación procedural. A partir de dicho estudio presentamos un prototipo que permite controlar la generación procedural de plantas de una forma sencilla, a la vez que versátil, mediante un dispositivo multitáctil.

1.2 Sistemas Multitáctil

Uno de los objetivos de la industria informática es desarrollar tecnologías amigables, innovadoras y atractivas para el usuario, que faciliten su uso y aprendizaje. La mayoría de periféricos tradicionales de entrada y salida (teclado, ratón, Trackball, cámara web, monitor, proyector, etc.) han evolucionado sorprendentemente poco en los últimos años. La interacción basada en un dispositivo apuntador, un teclado y un monitor ha variado poco en sus más de 35 años de edad. Una de las tendencias tecnológicas más recientes, que ha despertado interés en la comunidad investigadora y ha empezado a introducirse en el mercado de consumo, son las interfaces multitáctiles. Dichas interfaces permiten atender a más de un punto de contacto a la vez, y funcionan como un periférico de entrada y salida, sustituyendo a los tres dispositivos principales que componen un ordenador moderno: el monitor, el ratón y el teclado.

Las interfaces multitáctiles se pueden encontrar hoy día en los mercados de telefonía móvil [1], pantallas de ordenador [2], bancos de trabajo [3, 4] e incluso en pizarras o pantallas de proyección [5]. Sin embargo, siguen siendo caras y difíciles de construir. Recientemente han surgido propuestas innovadoras para implementar interfaces multitáctiles con componentes comunes [6] así como software y librerías para facilitar su programación [7].

Los componentes principales de las pantallas multitáctiles actuales son un proyector, una cámara web y una superficie de proyección. Por lo general este tipo de pantallas necesitan bastante espacio para situar todos los componentes, ya que para mostrar una imagen, el proyector necesita una distancia mínima. Dicho proyector, además de ser costoso, frágil, y por lo general pesado, necesita un mantenimiento regular, debido a lo reducido de la vida útil de su fuente de luz.

Hay diferentes tipos de tecnologías para la captura de la información táctil [1, 8], siendo las cámaras infrarrojas la tecnología más usada, más fácil de implementar y de menor coste. Sin embargo, estas cámaras tienen limitaciones como su ángulo de visión y la necesidad de situarlas a una distancia mínima de la superficie de proyección.

Por otro lado, el uso de tecnologías alternativas y nuevos desarrollos de la industria permiten resolver dichos problemas. Usar una pantalla LCD reduce considerablemente las dimensiones necesarias para obtener una imagen. Por otro lado, recientemente se pueden encontrar en el mercado cámaras web con lentes de gran angular, lo que permite reducir el espacio necesario para la captura de la superficie de contacto. Como se expondrá más adelante, la utilización de un espejo permitirá reducir aún más el tamaño del dispositivo.

1.3 Generación Procedural

Los juegos de rol multi-jugador masivos en línea o MMORPG (massively multiplayer online role-playing games) modelan grandes entornos para que gran número de jugadores puedan compartir experiencias. Para obtener paisajes realistas y campos de juego interesantes, los juegos modernos necesitan una gran cantidad de modelos geométricos y texturas complejas. Esto representa un problema, ya que es muy costoso crear manualmente diferentes objetos, paisajes, personajes, etc. Cambiar ciertas características de un objeto (por ejemplo el color, su tamaño, o su textura) es una técnica habitual para incrementar el número aparente de objetos distintos en el juego. Sin embargo, esta técnica es usualmente detectada por el usuario, lo que reduce el realismo del juego.

Los recursos dedicados a la creación de modelos realistas se podrían usar para mejorar el modo de juego o incluir características innovadoras. Las técnicas de generación de contenido procedural incrementan la velocidad de proceso de creación de contenido. También son capaces de generar contenido bajo demanda, reduciendo los requisitos de espacio de almacenamiento.

Recientemente, los sistemas de generación automática de contenido se han usado, por ejemplo, para modelar edificios y ciudades [9, 10], caminos [11], edificios [12,13], casas [14], texturas [15], vegetación [16,17] y cielos [18]. Específicamente para juegos, los sistemas procedurales también se han usado para modelar mapas 2D [19] y niveles de juego [20].

Nosotros nos concentramos en el requisito de los juegos de presentar vegetación realista. Este es un objetivo difícil porque las plantas naturales son organismos complejos y hay que tener en cuenta múltiple factores para definir su forma y color. Es posible modelar plantas realistas con los métodos tradicionales, pero usualmente resulta en modelos geométricos muy detallados, con bastantes texturas, lo que requiere un esfuerzo grande de modelado.

La visualización de modelos complejos también suele necesitar la aplicación de técnicas de LOD para reducir el número total de polígonos procesados en la escena. Algunos algoritmos de generación procedural son capaces de generar modelos de muti-resolución [17] y otros de generar plantas basadas en imágenes [21].

1.4 Estructura del trabajo

El resto de este trabajo está estructurado como se explica a continuación. Primero se expondrá el estado del arte, en el que se presentará la definición de las interfaces multitáctiles y de la generación procedural, sus tecnologías y tipos más comunes y sus trabajos previos.

El trabajo continúa con una sección de análisis del trabajo, donde se explica la problemática, la justificación, los objetivos, así como los alcances y limitaciones del mismo.

La siguiente sección presenta la implementación de la interfaz multitáctil en dos etapas, un banco de trabajo multitáctil y un sistema multitáctil portable.

A continuación se presentará el estudio realizado sobre los generadores procedurales disponibles actualmente para la creación de árboles. En esta sección también se presentará la implementación de un software de generación procedural que permite la interacción multitáctil, aplicando ciertas características de los sistemas evaluados.

Las secciones finales hablan de los resultados obtenidos en cada una de las fases, así como de las conclusiones y trabajo futuro. Se incluye una sección donde se presentan las publicaciones logradas y la bibliografía. Se finaliza con los anexos, donde se incluye un glosario corto de términos importantes utilizados en el trabajo y se adjuntan los artículos publicados relacionados con esta tesis.

2 Estado del Arte

2.1 Interfaces Multitáctiles

La tecnología táctil actual, como la utilizada en PDAs, tabletas gráficas, cajeros automáticos, terminales de punto de venta y algunos aparatos de telefonía móvil presentan el inconveniente de que solo permiten procesar un punto de contacto a la vez.

La tecnología multitáctil es la tecnología interactiva que permite a los usuarios controlar aplicaciones gráficas con varios dedos. Hay varios tipos de sensores para construir interfaces multitáctiles. Algunos de estos sensores capturan proximidad, acústica, capacitancia, resistividad, movimiento, orientación y presión, aunque, como se verá a continuación, los sensores más utilizados son los ópticos.

2.2 Tecnologías Multitáctiles

La arquitectura multitáctil está formada normalmente por cuatro componentes: la pantalla, el proyector, la distribución infrarroja y la cámara infrarroja. Dependiendo de cómo se combinen estos elementos, encontramos distintos tipos de implementaciones, tal y como se ve a continuación.

Las cinco tecnologías para la creación de sistemas multitáctiles más usadas en la actualidad son: Frustración de reflexión total interna (FTIR) [6], Iluminación Difusa (DI) [2], Plano de Luces Laser (LLP) en investigación por Microsoft, Iluminación Difusa de Superficie (DSI) y Plano de Luz LED (LED-LP). Las tres últimas se pueden encontrar en [7].

También hay otros tipos de sensores para construir interfaces multitáctiles. Algunos de estos sensores capturan proximidad, acústica, capacitancia, resistividad, movimiento, orientación y presión.

2.2.1 Reflexión Interna Total Frustrada (FTIR).

La Reflexión Interna Total (TIR) es un fenómeno óptico que se produce cuando un rayo intenta pasar de un material a otro con menor índice refractivo y con un ángulo específico de incidencia respecto a la normal de la superficie que hace que el rayo se refleje internamente. El ángulo específico con que ocurre esto, depende de los índices de refracción de ambos materiales, y es conocido como el ángulo crítico, que puede ser calculado matemáticamente usando la ley de Snell. Utilizando este principio en un acrílico con leds infrarrojos a sus costados, se produce la reflexión total interna de luz infrarroja. De este modo, la luz se encuentra rebotando dentro del acrílico. Al tocar el usuario la superficie del acrílico, se dice que la reflexión se frustra, permitiendo que los rayos de infrarrojo se escapen del acrílico, y sean capturados por la cámara.

Un sistema multitáctil basado en FTIR, está constituido por una superficie de proyección, una superficie de contacto (generalmente silicona transparente) y un acrílico transparente donde se produce el efecto de TIR (con un marco en la orilla para evitar que la cámara capture los infrarrojos que escapan al llevar un ángulo mayor al necesario), un proyector y una cámara de infrarrojos (véase la Figura 1). La superficie de contacto es necesaria para ayudar a la superficie de proyección a frustrar la luz en el acrílico, además de para ayudar a mejorar la sensibilidad de la captura.

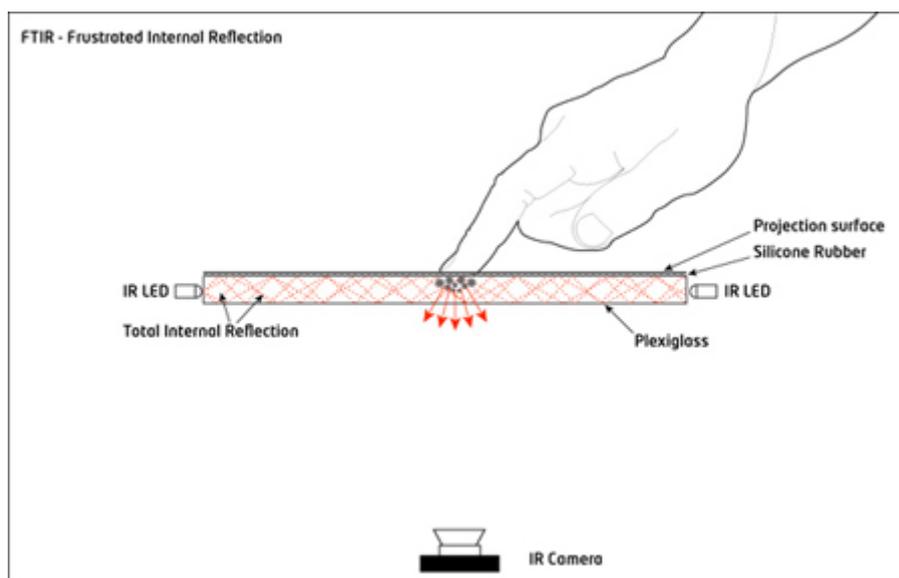


Figura 1: Estructura de una pantalla con Reflexión Interna Total Frustrada.

Una alternativa a este formato consiste en sustituir el proyector y la superficie de proyección por una pantalla LCD. En general, este tipo de técnica tiene estos problemas: La técnica FTIR necesita de una superficie de contacto si se desea contar con un reconocimiento de un toque en movimiento, debido a que si no la tiene no se reconoce claramente si los dedos están secos. Por el momento, no se ha logrado crear una superficie de contacto clara y libre de distorsiones para el LCD, que ayude a la sensibilidad. Por otro lado los sistemas multitáctiles LCD-FTIR siguen ocupando bastante espacio y la iluminación no es tan buena.

2.2.2 Iluminación Difusa (DI).

Este tipo de arquitectura sólo necesita una superficie de proyección que deje pasar luz infrarroja, por lo general un acrílico blanco (véase la Figura 2). Es bastante complejo realizar una distribución uniforme de la luz infrarroja en este tipo de arquitectura, por lo que a menudo resulta que ciertas partes de la pantalla reciben más luz que otras.

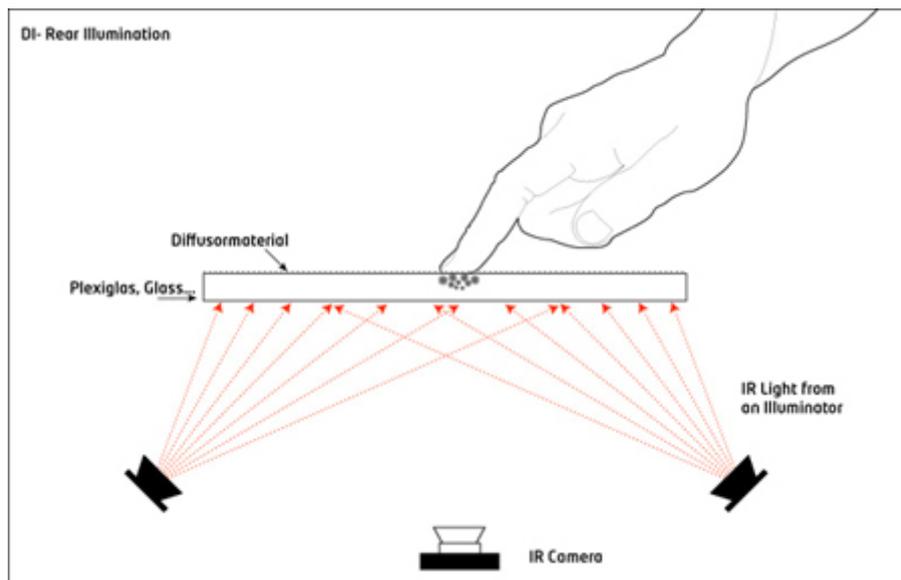


Figura 2: Estructura de una pantalla con iluminación Difusa Trasera.

Dependiendo de la posición relativa entre la luz, la cámara y la pantalla, existen dos tipos de iluminación difusa: frontal y trasera. La iluminación frontal funciona capturando las sombras que se producen cuando el usuario toca la pantalla. La iluminación trasera, por otro lado, provoca que el dedo refleje la luz infrarroja, y es ese reflejo el capturado por la cámara. La iluminación trasera necesita algún tipo de difusor para distribuir la luz, y frecuentemente este difusor funciona también como superficie de proyección. Estas técnicas tienen como ventaja poder reconocer figuras

en el punto de contacto (y no simplemente un blob). Esto permite usar dispositivos marcadores físicos con marcas, que codifican alguna acción predeterminada [23], como se puede ver en la Figura 3. La desventaja de esta tecnología es que no funciona bien en lugares iluminados.

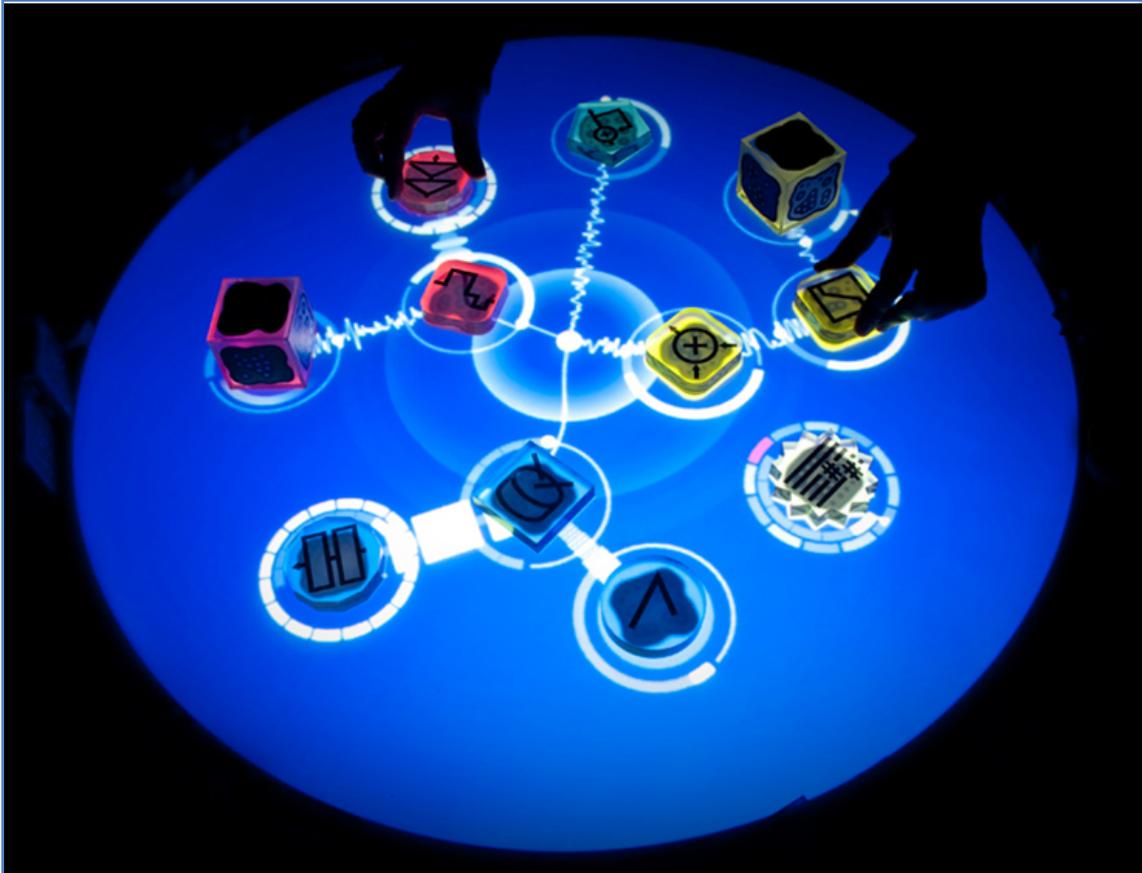


Figura 3: The Reactable.

2.2.3 Plano de Luz Laser (LLP).

Esta tecnología necesita una superficie de proyección que deje pasar luz infrarroja. Se dispone una malla de luz láser que abarca toda la superficie por el lado donde se encuentra el usuario. Al romper el usuario el plano, hay luz que es rebotada hacia abajo y una cámara detecta la luz infrarroja producida. El gran problema de utilizar luz láser es la seguridad del usuario, por lo que es necesario protección especial para los ojos (véase la Figura 4).

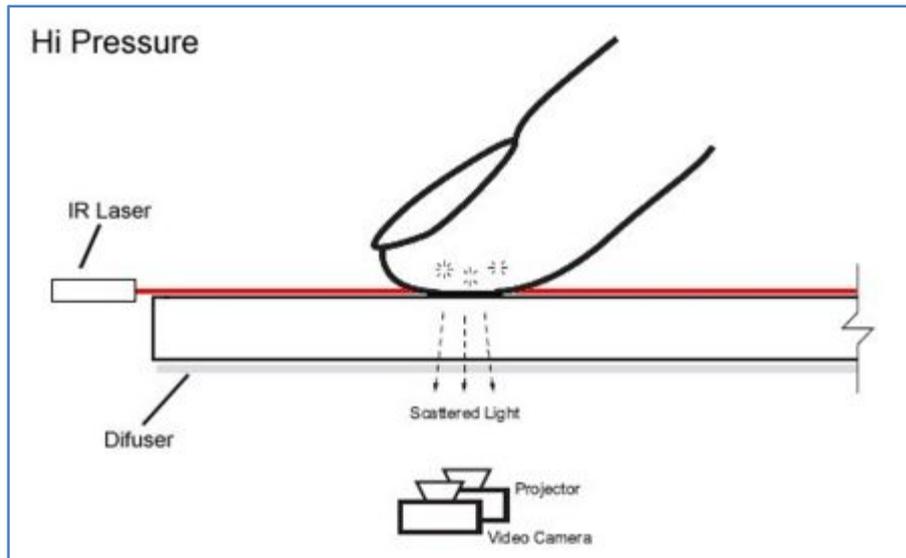


Figura 4: Estructura de una pantalla con Plano de Luz Laser.

2.2.4 Iluminación Difusa de Superficie (DSI).

Similar a la arquitectura FTIR, utiliza LEDs infrarrojos situados a un lado de un acrílico especial. Su funcionamiento es como un DI, como muestra la Figura 5. Dicho acrílico tiene partículas dentro del material que actúan como miles de pequeños espejos. Al irradiar con luz infrarroja en el borde del acrílico, es distribuida y lanzada en las dos direcciones.

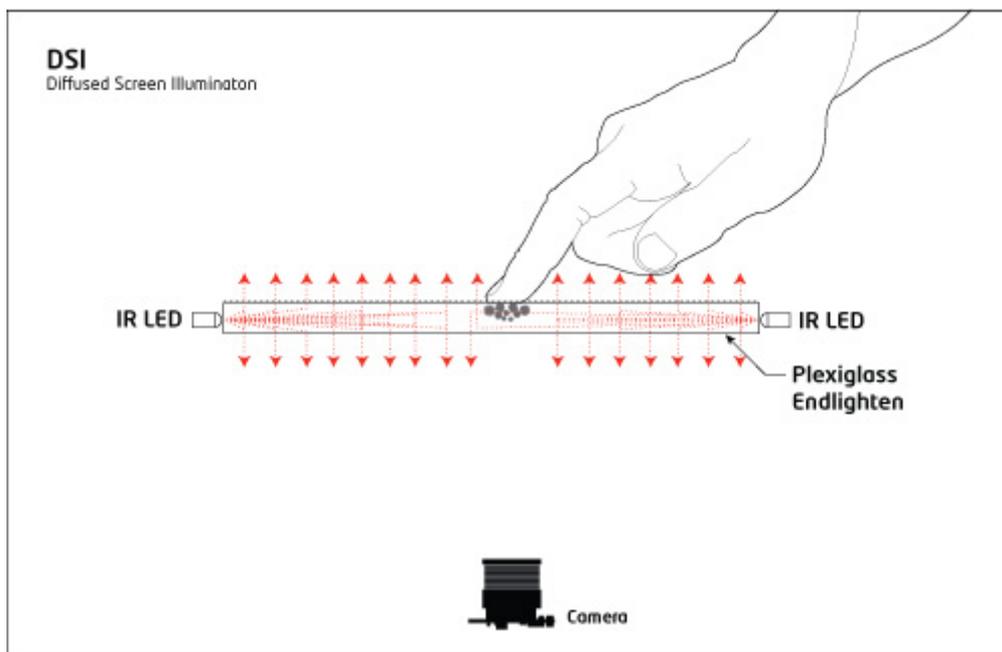


Figura 5: Estructura de una pantalla con Iluminación Difusa de Superficie.

2.2.5 Plano de Luz LED (LED-LP)

La arquitectura es similar al FTIR-LCD. La diferencia principal es que se retira el acrílico que produce el efecto de TIR y así se crea el plano de luz. Es necesario poner un borde extra a los LEDs para evitar que los rayos de luz salgan con un ángulo excesivo y sean captados por la cámara lejos de la pantalla (ver Figura 6). Esta técnica solo es útil en arquitecturas con pantallas LCD, ya que con proyectores dan mejores resultados las técnicas FTIR y DI.

Los LEDs infrarrojos son posicionados alrededor de la superficie, rodeando por todos sus lados, similar a una distribución LLP. Crea un plano de luz infrarroja sobre la superficie. Como la luz de los LEDs es cónica, en vez de plana como en los laser, la luz alumbrará objetos que se encuentren arriba de la superficie en vez de estarla tocando. Esto puede ser ajustado mediante ajustes en el software.

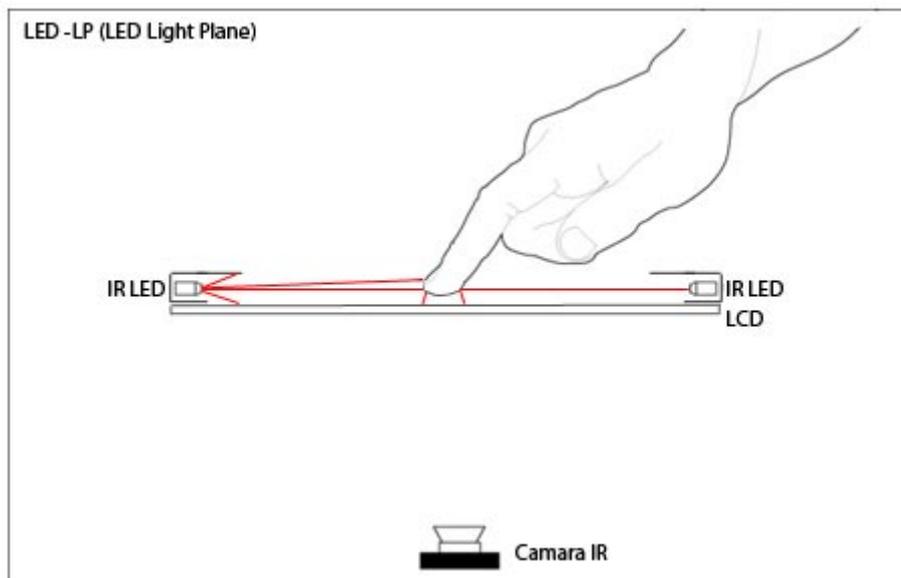


Figura 6: Estructura de una pantalla con un Plano de Luz LED.

2.3 Trabajos previos

Se pueden encontrar diversas aplicaciones en las que se han utilizado interfaces multitáctiles, como sintetizadores de música [22,23] (ver Figura 3 y Figura 7), juegos de mesa [24], difusión de noticias [25], paredes de escalada interactivas [5] o el manejo de aplicaciones 3D [26], como se puede ver en Figura 8.

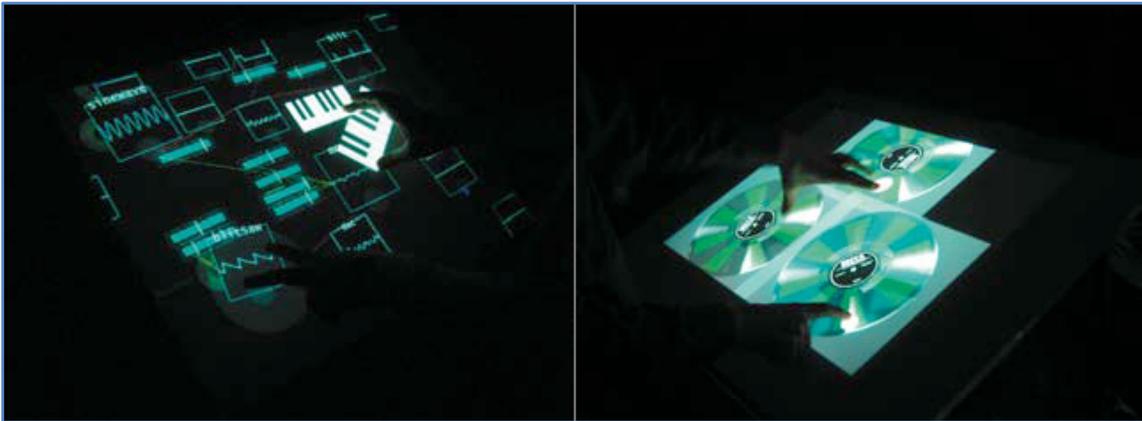


Figura 7: Sintetizador de Música.



Figura 8: The CubTile prototipe. Interfáce de manejo de aplicacions 3D.

Parejo al desarrollo hardware, han aparecido diferentes sistemas software en forma de aplicaciones finales o bibliotecas. Típicamente dicho software se desarrolla *ad hoc* para cada plataforma. Echtler y Klinker [27] proponen una arquitectura que intenta crear una aplicación base o estándar, unificando las similitudes y características del software existente.

También ha habido estudios sobre la percepción y control del usuario en una interfaz multitáctil usando una o dos manos [28], así como los gestos que se pueden hacer con toda la mano [29].

Una de las tareas típicas en un dispositivo multitáctil es simular un teclado virtual, aunque no siempre se cuenta con suficiente espacio para presentar un teclado completo. [30] propone una nueva técnica de uso de teclado reducido para pantallas multitáctiles donde sólo se requiere movimientos mínimos de dedo. Detectar una acción de clic en un dispositivo multitáctil presenta una dificultad añadida debido a la velocidad en la que se acerca y aleja el dedo. Para intentar resolver dicho problema se ha propuesto la combinación de tecnologías de reconocimiento de infrarrojos y de sombras [31], como se muestra en la Figura 9.

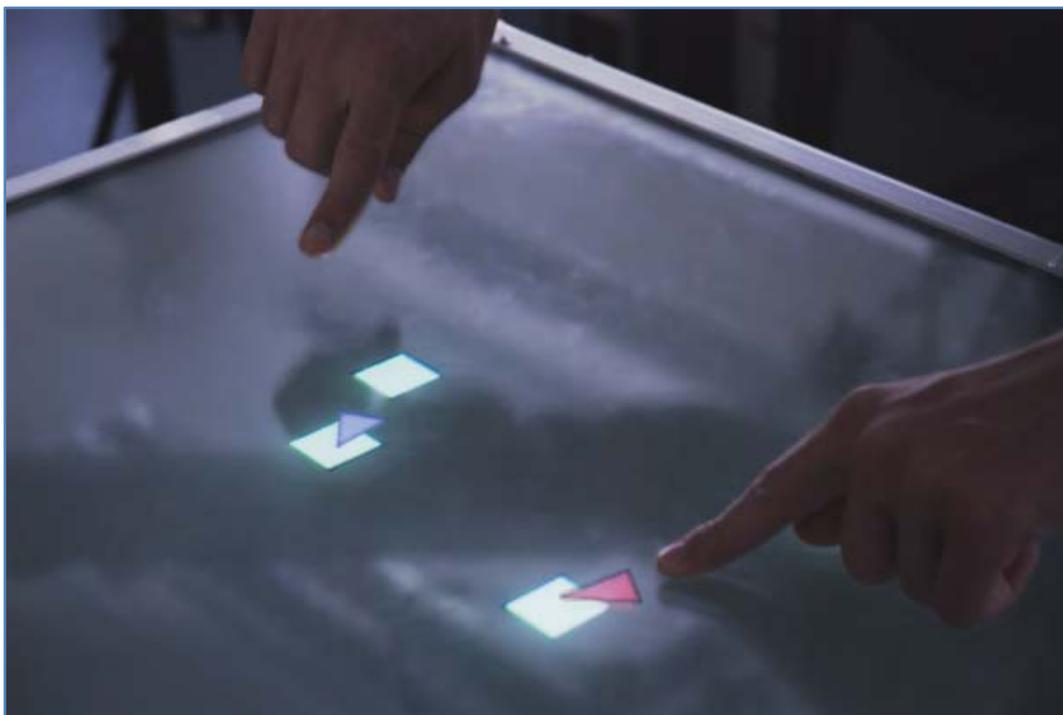


Figura 9: Reconocimiento de infrarrojo y sombras, el dedo tocando y el dedo cerca de la superficie.

Para reducir el espacio necesario para construir un dispositivo multitáctil, hay grupos que proponen utilizar varias cámaras [7]. El problema de dicha solución, aparte del aumento del coste del sistema es también el aumento del procesamiento a realizar cada vez que se genera una imagen. La solución propuesta en este trabajo utiliza una única cámara.

2.4 Generación Procedural

El término generación procedural se usa principalmente en la producción de medios, tales como modelos geométricos, imágenes o sonidos mediante la ejecución de programas o procedimientos. Por lo general se usa en aquellas aplicaciones en las que se necesita mostrar gran cantidad de modelos detallados. Un ejemplo es el diseño de videojuegos. Otra característica de este tipo de técnicas es que pueden generar contenido bajo demanda, reduciendo el tamaño de los recursos utilizados por las aplicaciones.

En los apartados siguientes describimos las principales técnicas de generación procedural utilizadas en información gráfica. Dentro de los generadores procedurales más comunes encontramos: los fractales, los sistemas L, las funciones de ruido y los algoritmos genéticos.

2.4.1 Fractales

Las primeras técnicas procedurales se basaron en funciones recursivas y se aplicaron a la creación de imágenes fractales. Un fractal es una figura geométrica fragmentada, donde cada fragmento es aproximadamente una copia reducida de toda la figura. No pueden ser clasificados en el modelo geométrico tradicional euclídeo, tienen una estructura rica en escalas pequeñas arbitrarias y su dimensión Hausdorff es mayor que su dimensión topológica. Algunas técnicas usan fractales para generar plantas, rocas [32] y otros fenómenos naturales [33].

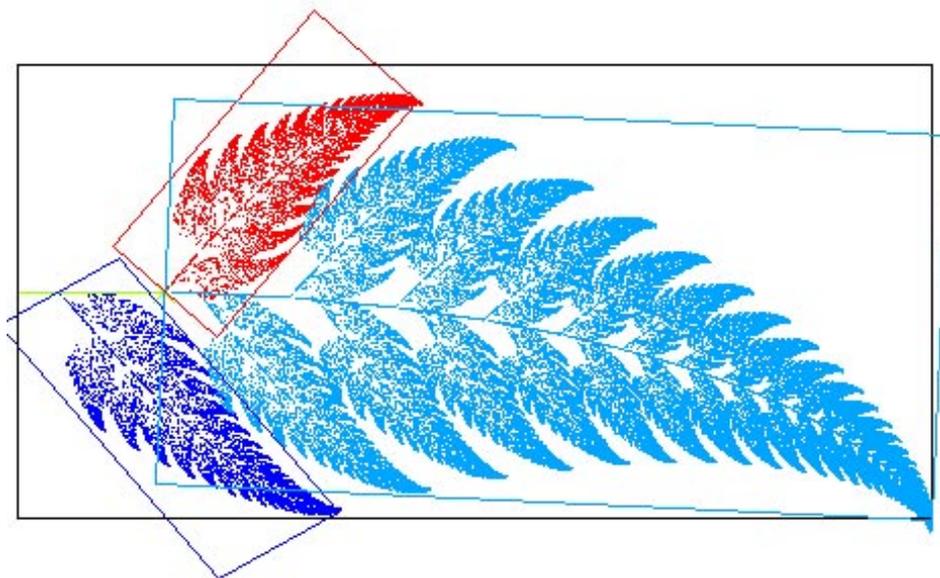


Figura 10: Una imagen Fractal.

2.4.2 Sistemas L

Otra técnica procedural son los sistemas-L [34]. Estos sistemas son una variación de las gramáticas formales y se usan para generar plantas, fractales o vida artificial.

Un sistema-L está formado por un conjunto de símbolos, un axioma o símbolo inicial y un conjunto de reglas de producción. El sistema-L comienza con el axioma, el cual es reemplazado por otros símbolos usando una regla de producción. Luego algunos de los nuevos símbolos son reemplazados usando otras reglas y así en adelante.

El resultado de este proceso es una cadena de símbolos que son interpretados por el visualizador como posiciones, orientaciones y estructuras de almacenaje. Los Sistemas L evolucionaron al incorporarse corchetes, los cuales significan ramificaciones o hijos. Este tipo de técnicas procedurales se usa para generar plantas complejas y otras estructuras naturales como árboles y conchas.

En el capítulo 5 se presentará un sistema-L en detalle.

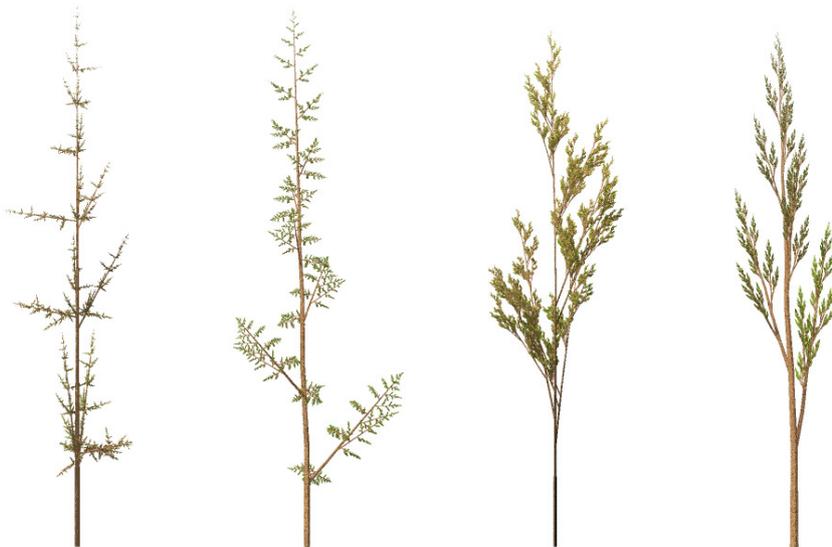


Figura 11: Árboles generados con Sistemas L.

2.4.3 Algoritmos genéticos

Este tipo de algoritmos se les llamó “genéticos” debido a su similitud con la evolución biológica y su base genético-molecular. Su funcionamiento consiste en hacer evolucionar una población de individuos, sometiéndoles a acciones aleatorias (mutaciones y recombinaciones genéticas). Luego se somete a los individuos a una selección de acuerdo a criterios de aptitud y en base a esa selección se deciden los que sobreviven y los que se descartan. Son métodos basados en probabilidades que

convergen a obtener el individuo óptimo. Su principal inconveniente es que suelen tener un alto coste.

2.4.4 Funciones de Ruido

Utilizando funciones de ruido, se pueden lograr texturas y paisajes muy realistas [15]. Además, se pueden generar texturas muy complejas, todo ello ocupando poco espacio. Estas técnicas se usan especialmente para la generación de texturas bajo demanda.

Una función de ruido, genera números aleatorios entre 0 y 1 al introducirle un número entero. Es importante que dicha función siempre genere el mismo número para que se pueda producir resultados. Usando una suave interpolación, se puede definir una función continua que toma un número no entero como parámetro. Si se toman varias de estas funciones con diferentes amplitudes y frecuencias, las sumas para crear una función de ruido. A esta función se le llama ruido de Perlin. Si a esto se le suma crear ruido en 2D, se crean texturas [15].

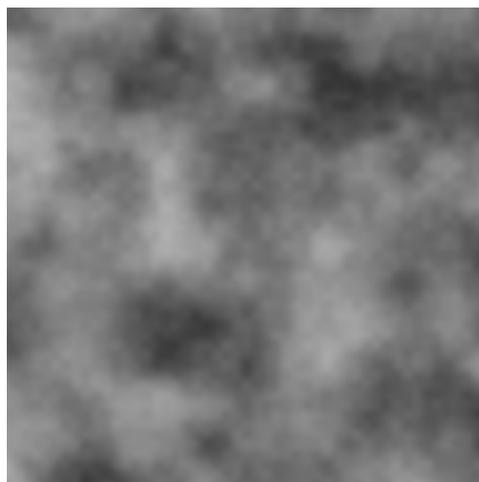


Figura 12: Textura a base de Funciones de Ruido.

2.5 Ejemplos de generadores procedurales

Recientemente han aparecido diversas técnicas procedurales para la generación de ciudades [35] (ver Figura 13). Podemos encontrar aplicaciones para generar ciudades en un terreno [9] (ver), organizar y simular ciudades con métodos procedurales [10], crear caminos, calles [11], áreas verdes y puentes, entre otros. También hay sistemas que modelan edificios [12], fachadas [13], casas [14]... Uno de los problemas principales que deben resolver este tipo de sistemas es la creación de diferentes edificios pero manteniendo un aspecto común (por ejemplo modelar diferentes edificios que se supone que se construyeron alrededor de la misma época, con el mismo estilo).

Kelly y McCabe [35] definen siete criterios para evaluar sistemas de generación procedural de ciudades: realismo, escala, variación de edificios y caminos, entrada requerida para generarla, eficiencia, control de modificación de lo generado y si es generado en tiempo real.

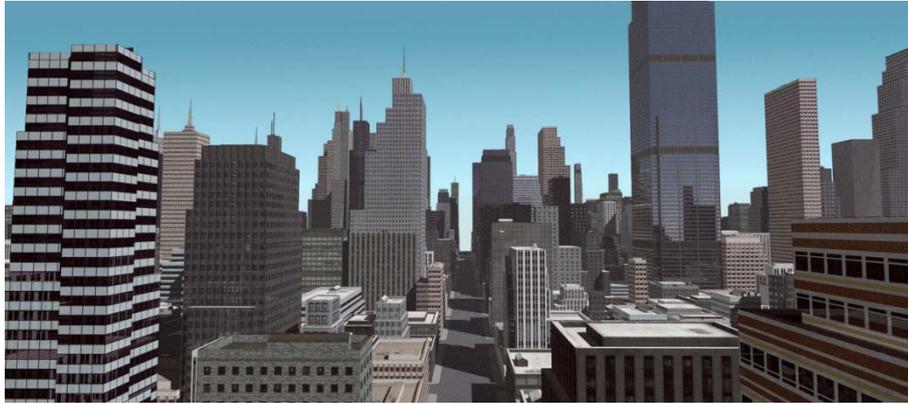


Figura 13: Modelado procedural de ciudades.

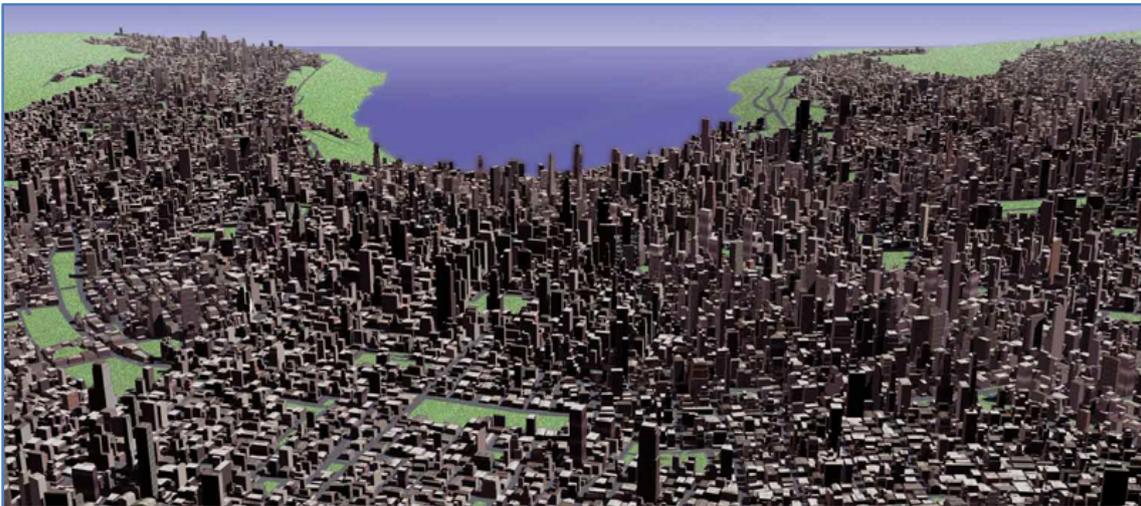


Figura 14: Modelado de una ciudad en un terreno.

Otros autores han presentado técnicas para el modelado de árboles reales [16], la reducción del número de polígonos del modelo de plantas utilizando técnicas procedurales [17], modelado de plantas basado en imágenes [21] (ver Figura 15) y animación de plantas [36].



Figura 15: Izquierda: Fotografía de una planta. Derecha: Modelo de planta basado en la Imagen.

3 Análisis

En la sección anterior se presentó el estado del arte sobre generación procedural e Interfaces multitáctiles. También se describieron las tecnologías multitáctiles más comunes, así como las técnicas procedurales más usadas.

En el presente capítulo se describe la problemática a la que se enfrenta este trabajo, su justificación, los objetivos que se espera cumplir, así como los alcances y limitaciones.

3.1 Descripción del problema

La generación procedural se enfrenta al gran problema de editar los modelos creados. Por lo general se crean los objetos siguiendo ciertas normas y algoritmos especificados por el usuario antes de la generación. Pocos sistemas permiten una edición interactiva de los objetos creados proceduralmente.

La creación y edición de objetos implica una gran cantidad de variables y controles, por lo que el usuario debe realizar un esfuerzo importante para aprender a utilizar los sistemas. Además, hay pocos sistemas que permitan controlar interactivamente el proceso de generación. Normalmente, este proceso se basa en la técnica de ensayo y error, es decir, se ajustan los parámetros, se genera el objeto y, si no es adecuado el resultado, se vuelven de nuevo a ajustar los parámetros.

Pensamos que las interfaces multitáctiles pueden aliviar los problemas anteriores, permitiendo a un usuario no experto controlar eficaz e interactivamente la generación procedural.

Para implementar la interfaz multitáctil, es necesario tener un sistema (tanto software como hardware) que permita capturar los múltiples contactos y procesarlos. Hoy día

no es fácil conseguir este tipo de herramientas en el mercado. También pensamos que sería deseable contar con un sistema portátil que facilitara su uso en un rango amplio de situaciones.

3.2 Justificación

La tecnología multitáctil se encuentra en pleno proceso de expansión y todo indica que pronto estará presente en muchos sistemas. Pero mientras la tecnología siga estando fuera del alcance del público general debido a su coste y lo aparatoso de los dispositivos, dichas aplicaciones seguirán quedando restringidas a entornos especializados.

Al crear un dispositivo multitáctil portátil de bajo coste, se eliminan dichos problemas. La propuesta de este trabajo es popularizar dicha tecnología para buscar nuevas áreas de aplicación, y estudiar su impacto en el desarrollo de interfaces gráficas de usuario avanzadas.

3.3 Objetivos

El objetivo principal de este trabajo es la creación de un dispositivo multitáctil portátil de bajo coste. Para demostrar su utilidad, se implementará una interfaz multitáctil interactiva sencilla para controlar una aplicación de generación procedural de imágenes.

Dentro de los objetivos específicos se encuentran:

- Crear una aplicación que mediante una interfaz multitáctil, permita al usuario crear fácilmente contenido procedural.
- Facilitar la creación procedural de estructuras y objetos (árboles) mediante una interfaz de usuario intuitiva y eficaz.
- Mejorar el control en la creación de un objeto generado proceduralmente.
- Facilitar a los usuarios el aprendizaje y la usabilidad de un sistema de generación procedural al utilizar tecnología multitáctil.
- Crear la arquitectura necesaria para trabajar con tecnología multitáctil.
- Mejorar la arquitectura multitáctil para su uso fuera del laboratorio, buscando ocupar el menor espacio posible.

3.4 Resultados y Limitaciones

Dentro de los resultados esperados se incluyen los siguientes:

- La pantalla multitáctil portátil será de bajo coste y ocupará poco espacio.
- La investigación de las tecnologías multitáctiles servirá para reconocer las necesidades para la interacción con el software a crear.
- El análisis de las características de los sistemas de generación procedural para modelar plantas, servirá para reconocer los requisitos necesarios para la creación del demostrador.
- La implementación del sistema de generación procedural para generación de árboles tendrá las características mínimas necesarias de uno de los sistemas evaluados.
- Para la implementación de la interacción con el sistema de generación procedural se utilizarán los prototipos creados y las librerías más adecuadas para su correcto funcionamiento.

En cuanto a las limitaciones, incluimos las siguientes:

- La calidad de construcción del dispositivo será limitada, debido a que es un prototipo. En un futuro, una vez validado el diseño, servirá para crear una versión definitiva.
- La interfaz del sistema no será muy compleja, debido a que no se busca crear un software comercial sino una prueba de concepto.

4 Implementación de interfaces multitáctiles

La parte principal de este trabajo consiste en la construcción de un dispositivo multitáctil. Los dispositivos disponibles comercialmente hoy día, o bien son demasiado caros [2], o bien son demasiado específicos [25].

En este capítulo se describe el proceso seguido para la construcción de dos sistemas multitáctiles: un banco de trabajo y un sistema portátil de bajo coste.

4.1 Banco de trabajo multitáctil

En la primera parte del desarrollo de este trabajo, colaboré en el proyecto BAnco de Trabajo como HErramienta para la Neurorehabilitación (BATHEN), dirigido por Francisco Abad, dentro del Instituto de Automática e Informática Industrial.

En dicho proyecto se construyó un banco de trabajo con una interfaz multitáctil. Para ello, se evaluaron los pros y contras de cada arquitectura, y finalmente se decidió utilizar la tecnología FTIR presentada en el capítulo 2. Como ya se comentó, los componentes principales de dicha tecnología son una pantalla de retroproyección, un acrílico, un proyector y una cámara de infrarrojos.

El primer paso para construir dicho banco, fue crear una pantalla FTIR. El componente principal de esta pantalla es el acrílico. Las dimensiones utilizadas son 100x72x0.8 cm.

La preparación del acrílico consistió en el lijado y pulido de los bordes para permitir la entrada de los rayos infrarrojos, emitidos por LEDs de infrarrojos.

Se creó un marco para sostener el acrílico formado por un carril de aluminio en forma de “u”, donde se realizaron perforaciones para introducir los LEDs.



Figura 16: Izquierda. Lijado del perfil del acrílico. Derecha. Creación del marco de madera.

El acrílico se ilumina únicamente por sus dos lados más largos. Se instaló un conjunto de 22 LEDs infrarrojos por lado, con sus respectivas resistencias (ver Figura 17 y Figura 18). Los LEDs están alimentados por una fuente de 12 V.

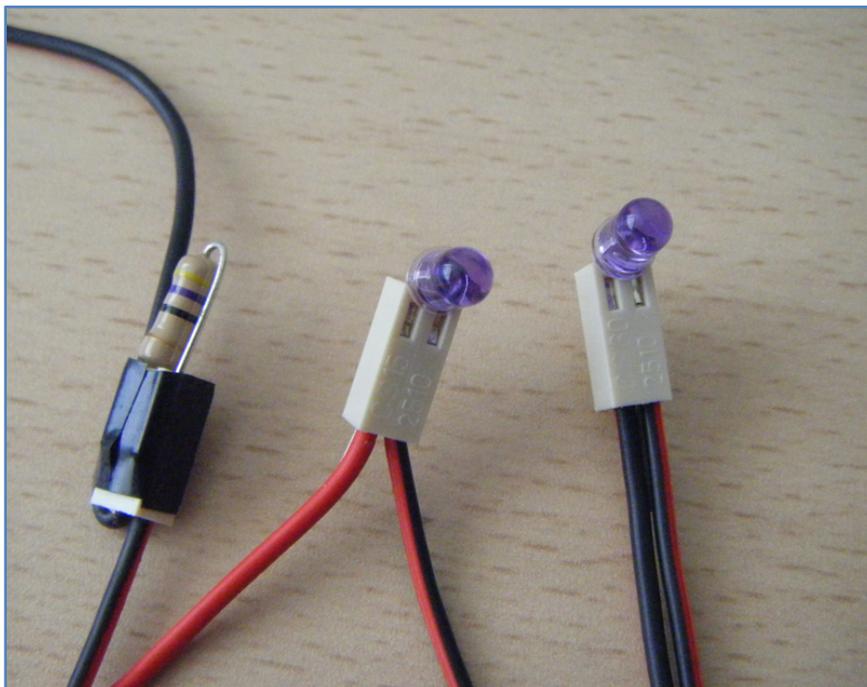


Figura 17: Armado del cableado, leds infrarrojos y resistencias.



Figura 18: Cableado y leds infrarrojos ubicados a los dos lados del acrílico.

Para la superficie de proyección, se utilizó una pantalla de retroproyección de la marca Rosco, de color gris, la cual ayuda a bloquear los rayos infrarrojos.

Se comprobó que si se dejaba la pantalla de retroproyección sobre el acrílico, el usuario tenía que presionar con fuerza sobre la pantalla para que la cámara pudiera captar el *blob* infrarrojo. Además, era muy difícil seguir los movimientos de los dedos sobre la pantalla.

Para resolver este problema, al utilizar una pantalla de retroproyección, es necesario crear una capa delgada de silicona para mejorar el contacto entre la pantalla y el acrílico. Se utilizó SORTA CLEAR 40 para crear una fina capa de silicona transparente (ver Figura 19), la cual una vez seca, creó un buen efecto para distinguir los blobs del lugar donde se tocara.



Figura 19: Izquierda. Aplicación de la capa de silicón. Derecha. Primer prueba.

Por último, se construyó un marco de madera para proteger la parte electrónica y el marco de aluminio.

Una vez que la pantalla fue funcional, se creó una mesa que pudiera sostenerla. Para ello, se tuvieron en cuenta los requisitos impuestos por el proyecto bajo el que se construyó. La mesa debía ser robusta y con la altura suficiente para que un usuario pudiera sentarse cerca frente a ella, con las piernas debajo de la mesa.

También se tuvieron en cuenta las distancias necesarias para situar el proyector y la cámara bajo la mesa. En el prototipo se utilizó un proyector marca Benq modelo MP522 ST. Dicho proyector tiene la óptica necesaria para proyectar una imagen de un metro de ancho a una distancia de 0.85m.

La cámara utilizada para capturar la imagen de la superficie de proyección es de la marca Philips, modelo SPC1001NC/00. Dicha cámara tiene 80 grados de ángulo de visión en horizontal y captura 60 imágenes por segundo.

Para permitir que la cámara capture la luz infrarroja, se le quitó el filtro de infrarrojos que incorpora la mayoría de cámaras de consumo (ver Figura 20) y se le instaló un filtro paso alto de 850nm (ver Figura 21).



Figura 20: Retirada del filtro infrarrojo.



Figura 21: Filtro paso alto de 850nm montado.

Una vez terminado el banco de trabajo, ha quedado funcional y listo para utilizarse (ver Figura 22, Figura 23 y Figura 24). Uno de los inconvenientes de la mesa es que es difícil de manejar en una habitación con mucha luz. Se están estudiando métodos para reducir esta dependencia.

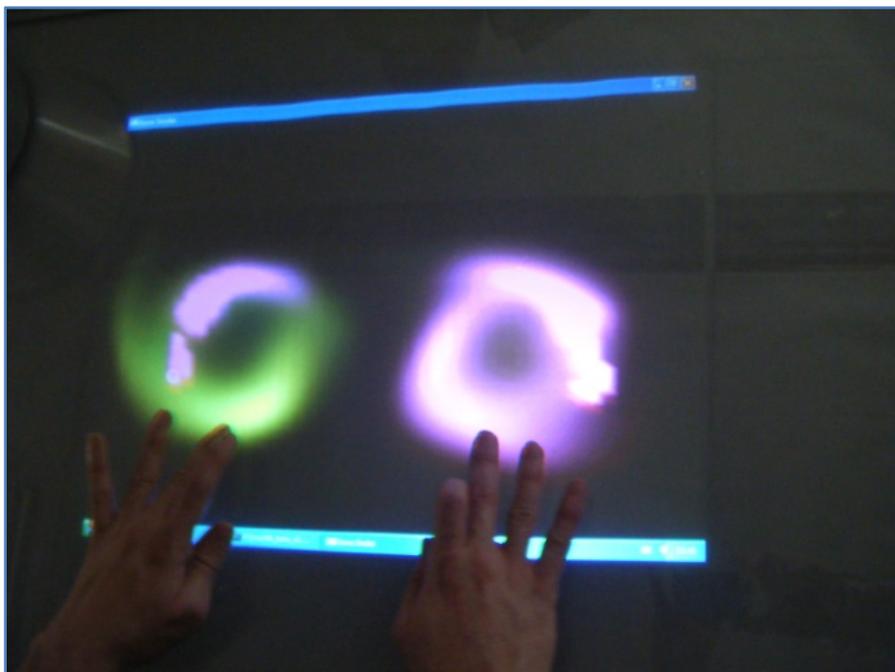


Figura 22: Mesa multitáctil en funcionamiento.

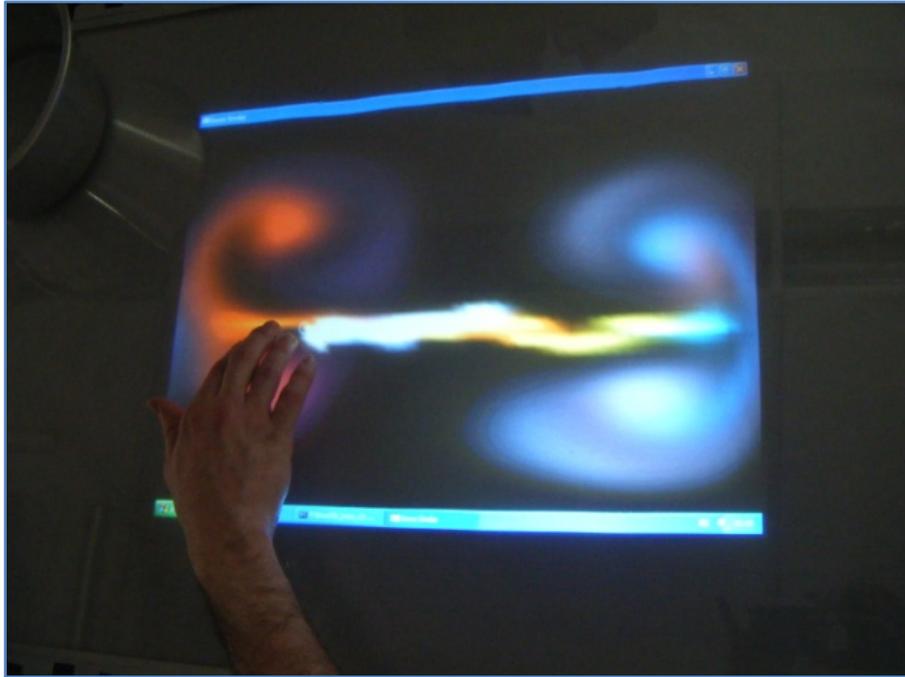


Figura 23: Pruebas de sensibilidad en la mesa multitáctil.



Figura 24: Banco de trabajo multitáctil finalizado.

4.2 Sistema multitáctil portátil

Con el primer objetivo logrado y con la experiencia adquirida, se comenzó a trabajar en el sistema portátil de bajo coste. Con el objetivo de construir una pantalla multitáctil orientada a su uso en casa, se plantearon los siguientes requisitos:

- ocupar poco espacio,
- ser portátil, por lo que además, debía pesar poco y
- ser barata.

Se descartó el uso de un proyector debido a dos limitaciones: son costosos y necesitan bastante espacio de proyección, incluso utilizando espejos. En vez de usar un proyector, se usó el panel de un monitor LCD, que elimina por completo la necesidad del espacio de proyección. Un monitor plano está compuesto básicamente por una pantalla semitransparente donde se crean las imágenes y por una fuente de iluminación trasera. La pantalla se basa en la tecnología LCD, que permite controlar píxel a píxel la cantidad de luz que pasa a su través, y por lo tanto, la intensidad del color percibida por el usuario.

Sin embargo, uno de los inconvenientes de las pantallas LCD es su fragilidad, lo que nos llevó a extremar el cuidado en el diseño del chasis del dispositivo para evitar su rotura durante su transporte. La Figura 25 muestra un esquema general con los elementos principales de nuestro sistema.

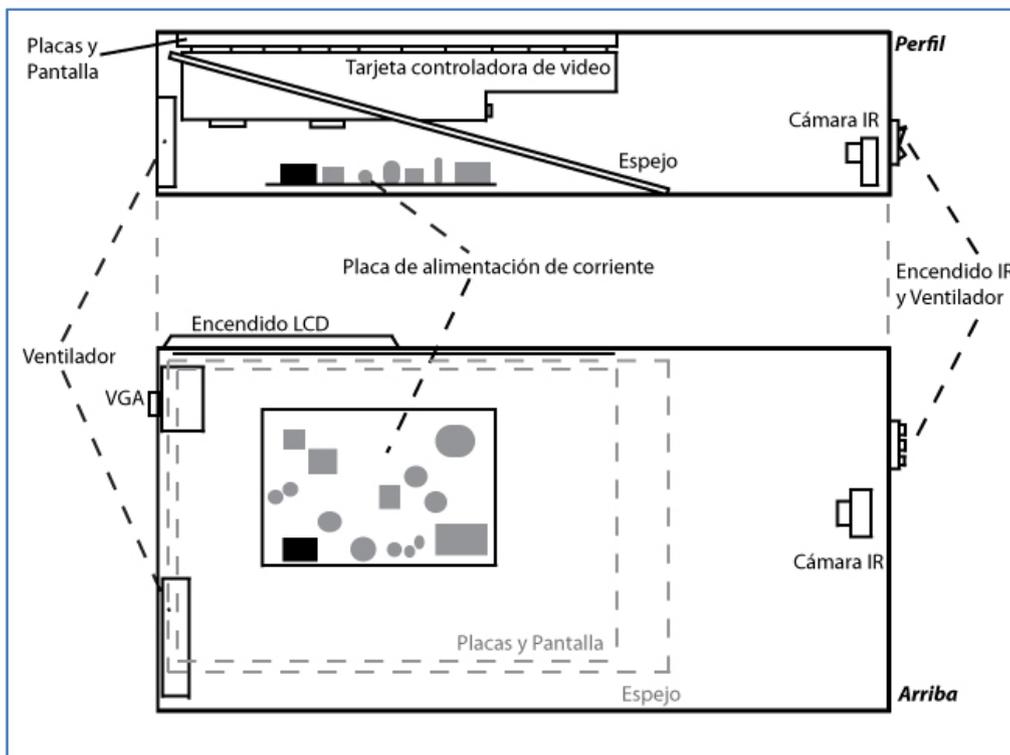


Figura 25: Diseño básico de la pantalla.

Se utilizó un monitor marca BENQ modelo FP71G de 17 pulgadas. En la Figura 26 se muestra el monitor desmontado.

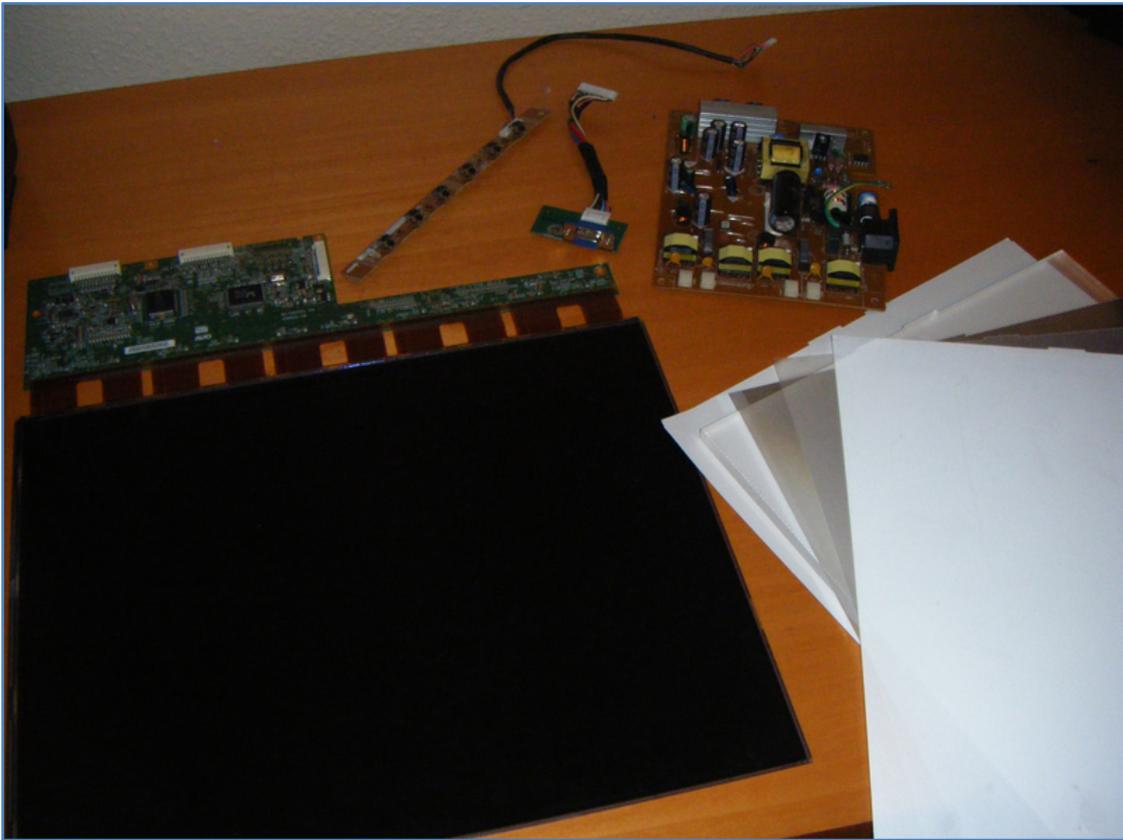


Figura 26: Monitor desarmado.

El ángulo de visión de las cámaras impone una distancia mínima para abarcar el objetivo. Sin embargo, recientemente han aparecido webcams con lentes de gran angular que reducen la distancia necesaria para cubrir completamente la pantalla de proyección. En nuestro diseño se utilizó de nuevo una cámara PHILIPS SPC1001NC/00, con las mismas modificaciones que el banco de trabajo (sin filtro infrarrojo, y con filtro paso alto de 850nm).

Los dispositivos multitáctiles de los que se ha tenido conocimiento y que utilizan la tecnología LCD toman normalmente la forma de una mesa, con la iluminación situada en el suelo, paralela a la pantalla. El resto de componentes se distribuyen para bloquear la menor cantidad de luz posible [7]. El problema de este diseño es que típicamente necesita una altura considerable.

Nuestro diseño, por otra parte, usa un espejo para reducir la altura del dispositivo, lo que introduce un problema a la hora de iluminar la pantalla. Se utilizaron reflectores con apertura de 40 grados, y fuentes de luz de 12 V de color blanco día. Posicionando las luces al lado de la cámara y apuntando al espejo, se obtenía una buena iluminación, pero insuficiente para que se distinguieran las imágenes generadas por el LCD. Esto era debido a que la luz no se distribuía uniformemente sobre la superficie de la pantalla y

se lograba ver el interior de la caja y sus componentes mediante el espejo. Este problema se resolvió usando un acrílico blanco detrás de la pantalla, lo que permite crear una iluminación uniforme.

Las dimensiones de la caja se calcularon en función del tamaño de la pantalla, el ángulo de visión de la cámara y su distancia mínima para abarcar el espejo.

Los LEDs infrarrojos utilizados son de marca OSRAM, modelo SFH485, cuyas especificaciones declaran que emiten luz de 880nm, a lo largo de 40 grados de emisión.

Una de las opciones que se barajaron al inicio del proyecto fue implementar la técnica de DI, aprovechando el espejo. Para ello, se colocaron dos reflectores con 12 LEDs infrarrojos cada uno, que apuntaban al espejo para abarcar toda la superficie de la pantalla. Al tocarla, ésta reflejaría los rayos infrarrojos en dirección contraria y la cámara los detectaría.

Esta técnica funcionaba únicamente sobre el acrílico blanco o sobre la pantalla LCD, pero no con los dos a la vez. La intensidad de los rayos infrarrojos era insuficiente para atravesar las dos placas (ida y vuelta) y que la cámara la detectara. Se intentó cambiar el acrílico por otro difusor de luz más delgado, o compuesto de otros materiales, pero se obtuvieron los mismos resultados.

Finalmente se optó por cambiar el método de detección de los contactos de DI a FTIR, lo que solucionó el problema. Para ello, se instaló un nuevo acrílico transparente de 3mm encima de la pantalla LCD, sin dejar espacio. Para proporcionar la iluminación infrarroja al acrílico, se utilizaron 12 LEDs infrarrojos colocados en serie a dos lados del mismo.

El siguiente problema que se tuvo que solucionar era el producido por las fuentes de luz que proporcionan la iluminación trasera al LCD. Dichas fuentes no producen luz infrarroja, pero producen reflejos tanto en el espejo como en la pantalla, reflejos que son capturados por la cámara de infrarrojos como blobs. Para solucionar el problema se diseñaron dos cajas difusoras, con dos fuentes cada una (véase la Figura 27). Sin embargo, la luz no era suficiente para que las imágenes se vieran completamente claras.

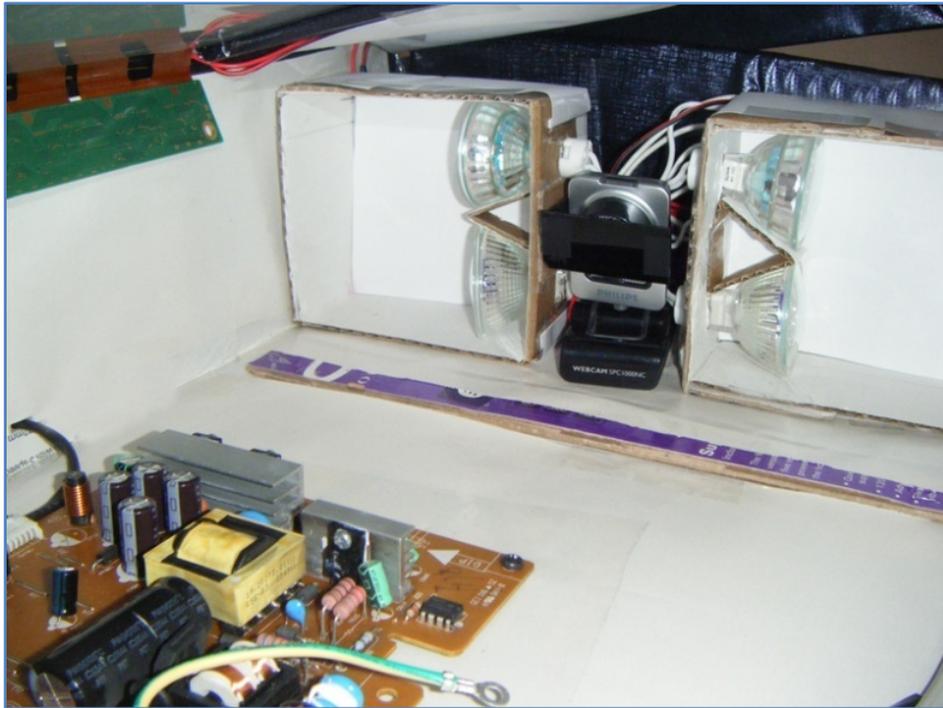


Figura 27: Iluminación trasera difusa.

El diseño final reutiliza la iluminación original del LCD. Dicha fuente de iluminación consiste en un acrílico transparente iluminado por tubos fluorescentes a sus costados, una lámina de plástico opaco en la parte trasera y tres láminas de distintos plásticos enfrente. En el monitor, el acrílico utiliza la técnica de TIR para distribuir uniformemente la luz sobre la pantalla. Originalmente, dicho acrílico está iluminado lateralmente. Una de sus caras cuenta con pequeños puntos marcados que rompen el efecto y permiten que la luz escape hacia el plástico opaco, en la parte trasera del monitor.

Las láminas delanteras son dos hojas difusoras y una hoja que filtra la iluminación infrarroja. Se retiraron el filtro de infrarrojos y el plástico opaco de la parte trasera, y se invirtió el sentido del acrílico. Los puntos marcados se orientaron hacia el LCD para dejar salir la luz, pero las hojas difusoras no fueron suficientes para evitar que se distinguieran los puntos en el LCD. Retirando las hojas difusoras y usando un acrílico blanco se consiguió una iluminación uniforme. La Figura 28 muestra la estructura definitiva de la pantalla.

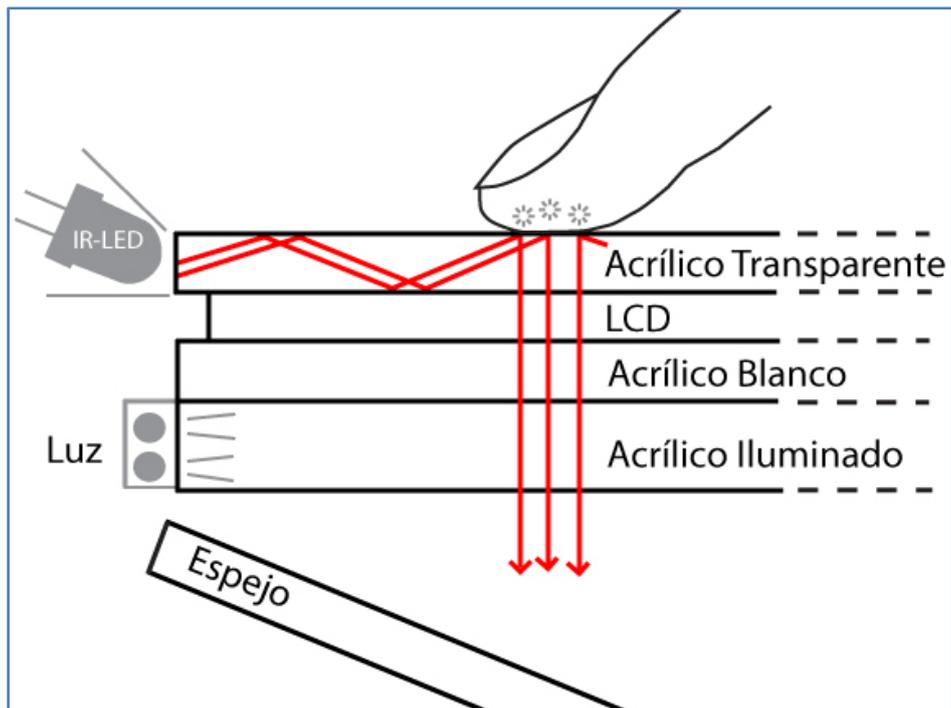


Figura 28: Estructura de las placas y el LCD.

Los cables planos flexibles (FFC, o Flat Flexible Cable por sus siglas en inglés) que van conectados a la placa controladora de video y a la pantalla no siempre permiten el uso multitáctil. En nuestro caso no se tuvo este problema, pero los cables al ser un poco cortos, provocan dificultades al conectarlos con la corriente y el conector VGA no permite abrir por completo la tapa mientras estén conectados, como se muestra en la Figura 29.

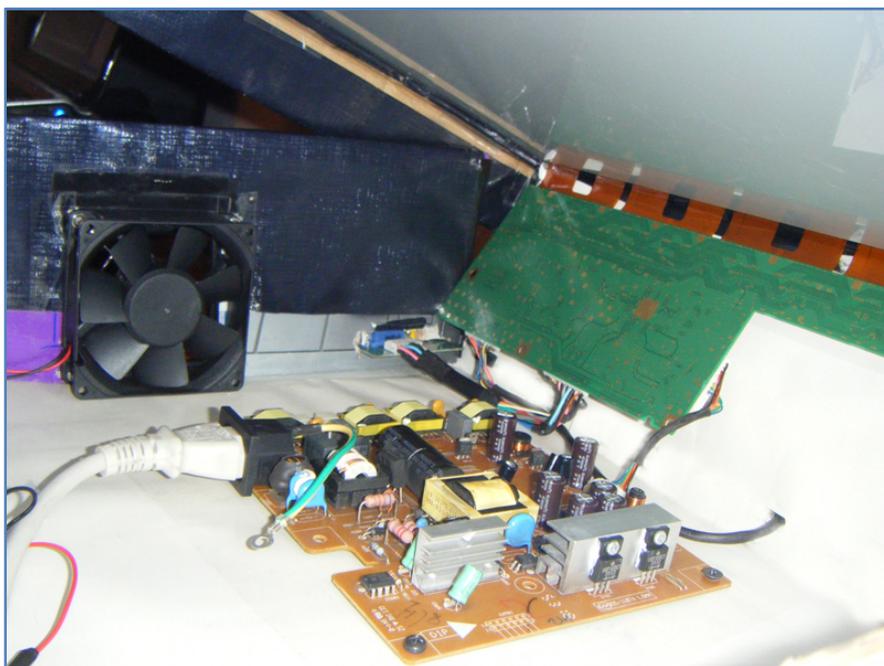


Figura 29: Cableado interno y problemas con extensión de cables.

Un ventilador en la caja evita el aumento de temperatura en el interior debido a la pantalla y su alimentación, la iluminación y las resistencias de los infrarrojos. También se han instalado dos interruptores para encender la iluminación infrarroja y el ventilador.

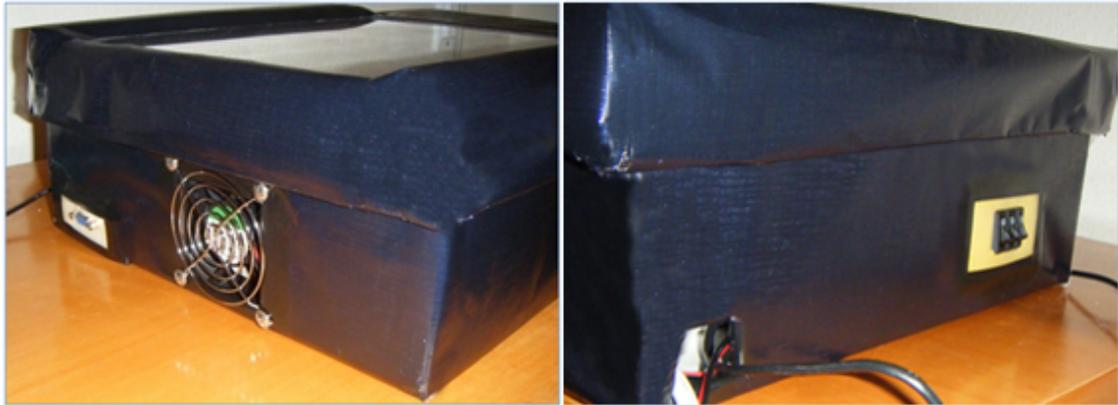


Figura 30: Detalles de la pantalla multitáctil. Ventilador, VGA e Interruptores.



Figura 31: La pantalla multitáctil terminada y funcionando.

5 Implementación de un generador procedural multitáctil

En este capítulo se describirá la aplicación que se utilizará para demostrar la interfaz multitáctil. Se seleccionó implementar una aplicación de generación procedural de plantas debido a la complejidad que presentan las aplicaciones de este tipo, y al conocimiento de dichas aplicaciones por parte del Grupo de Informática Gráfica.

5.1 Estudio Previo

Como se ha comentado en secciones anteriores, el primer paso para el diseño de una aplicación procedural multitáctil fue el estudio de los sistemas de generación procedural actuales para modelar plantas naturales. Se clasificaron dichas herramientas en diferentes grupos dependiendo de sus características y de los requerimientos del usuario. FractTree [37] y L-System4 [38] están orientados para su uso por estudiantes, debido a que ayudan al aprendizaje y comprensión de los sistemas L. Speedtree [39] y Xfrog [40] son recomendables para las compañías de películas y videojuegos, porque generan los árboles más realistas, pero también son los que tienen una curva de aprendizaje más lenta y tienen un coste mayor.

L-Studio [41] también crea flores y plantas realistas, pero requiere bastante tiempo de aprendizaje. An Ivy Generator [42] crea una escena u objeto con una apariencia abandonada o viva, debido a las plantas enredaderas que agrega. Al ser gratuita, tiene también la ventaja de que puede ser usado por todo el mundo. Treemagik G3 [43] y Tree[d] [44] generan resultados buenos y son baratos o gratuitos.

Algunos sistemas pueden ser usados para animaciones 3D no fotorrealistas de plantas y árboles. L-System4 [38], Dryad [45] y TreeGenerator [46] generan este tipo de árboles en vez de plantas realistas. Para el usuario casual FractTree [37], An Ivy Generator [42], Dryad [45], TreeGenerator [46], MeshTree studio [47] y Arbaro[48] proveen bastantes parámetros para seleccionar y crear diferentes resultados.

Trabajando con estas herramientas se constató la necesidad de un sistema que ayudara a la creación y edición del modelo de un árbol. Consideramos necesario estudiar nuevos sistemas de interacción, que faciliten el uso de aplicaciones complejas.

Para estudiar de forma más ordenada las aplicaciones, se creó una tabla comparativa con las características principales de cada sistema. Las características estudiadas son:

- 2D y 3D: Capacidad para generar objetos 2D y/o 3D.
- IDE: Editor integrado para crear las plantas.
- Lenguaje de Definición: Entrada para la definición de plantas. Puede ser tipo LSys[] (Sistemas L utilizando corchetes) o GUI (Interfaz Gráfica de Usuario).
- Navegación: Utilización de navegación en la escena generada (zoom, mover y rotar).
- Generación de geometría: Si puede generar geometría.
- Importar, Exportar: Formatos de ficheros aceptados por la aplicación.
- Liberado, Actualizado: Fecha de primera liberación y de última actualización.
- Control de derivación: Si se puede controlar la derivación de la creación de plantas.
- Objetos múltiples: Permite la creación de varios objetos a la vez.
- Tipo de objetos: Objetos con los que el sistema puede trabajar.
- Texturas: Capacidad de usar texturas en la creación de plantas.
- Propósito: Propósito principal de la aplicación.
- Usabilidad: Facilidad de uso en una escala del 1 al 10 (donde 1 es difícil y 10 fácil).
- Documentación: Calidad de la documentación en una escala del 0 al 10 (donde 0 es sin documentación, 1 documentación pobre, 10 completa y amigable).
- Depuración: Disponibilidad de herramientas para depurar el proceso de generación.
- Velocidad de modelado: Tiempo para crear una planta específica en cada sistema. Puede ser de un proyecto en blanco o modificando alguno ya existente. La planta debe tener estas especificaciones: una rama inicial (o tronco), una separación en dos ramificaciones, cinco derivaciones y hojas de redondeadas y tamaño medio.

	FracTree	L-System4	L-Studio Botany Alg	An Ivy Generator	TreeGenerator	TreeMagik G3
2D	x	-	x	-	-	-
3D	-	x	x	x	x	x
IDE	x	x	x	x	x	x
Definition Language	Lsys []	Lsys []	Mod. Lsys []	GUI	GUI	GUI
Navigation	-	x	x	x	x	x
Geometry Generation	-	x	x	x	x	x
Image Generation	-	x	x	x	x	x
Import	-	dxif	-	obj	-	b3d
Export	bmp	dxif, bmp, jpeg	rgb, tga, bmp	obj	obj, 3ds, dxf	several (obj, wrf)
License	Shareware	Freeware	Freeware	Freeware	1.3 & 2.0	Demo
Price	10 EUR	0	0	0	\$0 & \$49	\$49.95
Released	1993	2000	1999	2007	2006	2006
Updated	No	2004	2004	2008	No	No
Derivation control	x	x	-	x	x	x
Multiple objects	-	x	x	x	-	-
Object types	Lsystems	dxif, Lsystems	Plants Fractals	3D Objects	Trees	Trees
Textures	-	x	-	x	x	x
Purpose	Fractal trees	Lsystems	Plants	Ivy Plants	Trees	Trees
Usability	6	8	8	9	8	7
Documentation	6	2	10	4	6	0
Debugging	Level control	-	-	Level control	Instant update	Level control
Modeling Speed	1	1	2	3	4	5

Tabla 1: Características principales de las aplicaciones analizadas. Para la fila con leyenda “Modeling Speed”: 1 rápido pero utilizando ejemplos y necesidad de conocer sistemas L; 2, muy lento, requerimiento de modificar ejemplos existentes; 3, muy rápido, requerimiento de usar ejemplos existentes, 4, rápido para crear pero difícil para modificar; 6, rápido pero no sirve la exportación; 7, normal y 8, no se pudo probar.

	MeshTree Studio	Dryad	Arbaro	XFrog	SpeedTree	Tree[d]
2D	-	-	-	-	-	-
3D	x	x	x	x	x	x
IDE	x	x	x	x	x	x
Definition Language	GUI	GUI	GUI	GUI	GUI	GUI
Navigation	x	x	x	x	x	x
Geometry Generation	x	x	x	x	x	x
Image Generation	-	-	x	x	x	x
Import	-	-	xml	-	-	-
Export	.mesh	obj	obj, povray, dxf	png, jpg	several	x, b3d
License	Freeware	Freeware	Freeware	Trial, lite, full	Trial, full	Freeware
Price	0	0	0	\$300/\$400	\$8495,	0
Released	2007	2007	2003	1996	2002	2002
Updated	No	2008	2004	2002	2009	2008
Derivation control	x	-	x	x	x	-
Multiple objects	-	-	-	x	x	-
Object types	Trees	Trees	xml	Plants	Plants	Trees
Textures	x	-	-	x	x	x
Purpose	Trees	Trees	Trees	Plants	Plants	Trees
Usability	6	5	5	9		5
Documentation	0	0	6	10		0
Debugging	Level control	Instant update	Instant update	a lot		Instant update
Modeling Speed	7	5	6	2	8	5

Tabla 1 (continuación): Características principales de las aplicaciones analizadas. Para la fila con leyenda "Modeling Speed": 1 rápido pero utilizando ejemplos y necesidad de conocer sistemas L; 2, muy lento, requerimiento de modificar ejemplos existentes; 3, muy rápido, requerimiento de usar ejemplos existentes, 4, rápido para crear pero difícil para modificar; 6, rápido pero no sirve la exportación; 7, normal y 8, no se pudo probar.

5.2 Implementación de un sistema de generación procedural multitáctil

Tomando en cuenta la funcionalidad general de un sistema de generación procedural, y basándose en las características de las herramientas, se decidió crear un sistema de generación procedural de árboles. El sistema debía ser capaz de igualar la funcionalidad y además agregar la interactividad multitáctil a un sistema de este tipo. La herramienta debía facilitar la creación de contenido a los usuarios. Se debe tener en cuenta que se ha desarrollado una prueba de concepto sencilla, que no intenta competir con un sistema de generación procedural de plantas comercial. Se ha optado por implementar la funcionalidad básica del sistema descrito anteriormente más sencillo, que únicamente dibuja plantas 2D utilizando un sistema L.

Los sistemas-L están formados por un conjunto de símbolos, un axioma (o símbolo inicial) y un conjunto de reglas de producción. Cada símbolo del axioma es sustituido por su correspondiente regla de producción. Si algún símbolo no tiene una regla de producción, se mantiene sin modificaciones y se pasa al siguiente símbolo. Una vez terminado se vuelve a evaluar la producción obtenida, reemplazando los símbolos por sus correspondientes producciones. A continuación se muestra como funciona un sistema L sencillo y sus producciones finales en cada recursividad.

Símbolos: {F, G, +, -}

Axioma: ++F-G

Reglas:

F = E+G

G = E-F

Inicio: ++F-G

Recursividad 1: ++ E + G - E - F

Recursividad 2: ++ E + E - F - E - E + G

Recursividad 3: ++ E + E - E+G - E - E + E-F

El sistema-L que se ha implementado es muy sencillo: únicamente define los siguientes símbolos constantes:

- “F”: significa que se dibujará un tramo de línea, en la dirección actual.
- “+”: se realizará un giro en el sentido de las agujas del reloj.
- “-”: rotación en el sentido contrario al de las agujas del reloj. El ángulo de ambos giros es fijo y está predeterminado.

Los símbolos de corchete tienen un significado especial: permiten almacenar pilas de estados:

- “[“: se guardan la posición y la dirección actuales
- “]“: se restaura la última posición y dirección almacenadas por un “[“ y continúa con el dibujado a partir de dicha posición.

El sistema comienza la derivación construyendo una cadena que contiene únicamente el axioma. A continuación recorre la cadena tantas veces como indique el usuario, sustituyendo en cada recorrido las variables por su cadena de sustitución, conforme establecen las reglas. El resultado final consistirá en una cadena llamada Producción final, que contendrá un conjunto de símbolos constantes. Dicha cadena se interpreta a continuación para obtener su representación gráfica aplicando las operaciones descritas anteriormente. La Figura 32 muestra la interpretación gráfica de dos cadenas de símbolos.

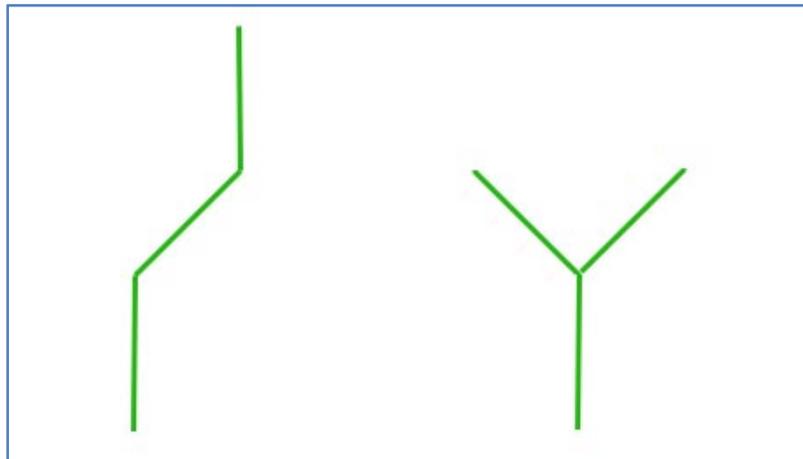


Figura 32: Una producción final de $F+F-F$ y una $F+[F]-[F]$ con un ángulo de 45 grados.

El sistema lee de un fichero de texto los parámetros necesarios del sistema L para generar un árbol. A continuación se describe el formato que deberán seguir dichos ficheros. El axioma y las reglas utilizadas, así como parte del funcionamiento fue basado en las características de FractTree [37].

Para comenzar, los comentarios comienzan por un punto y coma, el texto que se encuentre a su derecha será ignorado hasta encontrar un salto de línea. El fichero comienza con dos números enteros: el ángulo y la recursividad del sistema L.

El ángulo se utilizará para determinar la magnitud del giro realizado por cada operador “+” y “-“. Por otra parte, la recursividad determina el número máximo de derivaciones.

La siguiente línea del fichero contendrá el axioma. Las reglas completan la definición del fichero. Una regla está formada por un símbolo y su resultado, separado por un espacio. Cada regla está separada por un salto de línea. A continuación se muestran dos ejemplos de árboles y sus respectivas imágenes (ver Figura 33).

<pre> ;----- ;Simple plant ;----- 25 4 ---G G GFX[+G][-G] X X[-FFF][+FFF]FX </pre>	<pre> ;----- ;Nice tree ;----- 15 4 ---F F FF+[+F-F-F][-F+F+F] </pre>
--	---

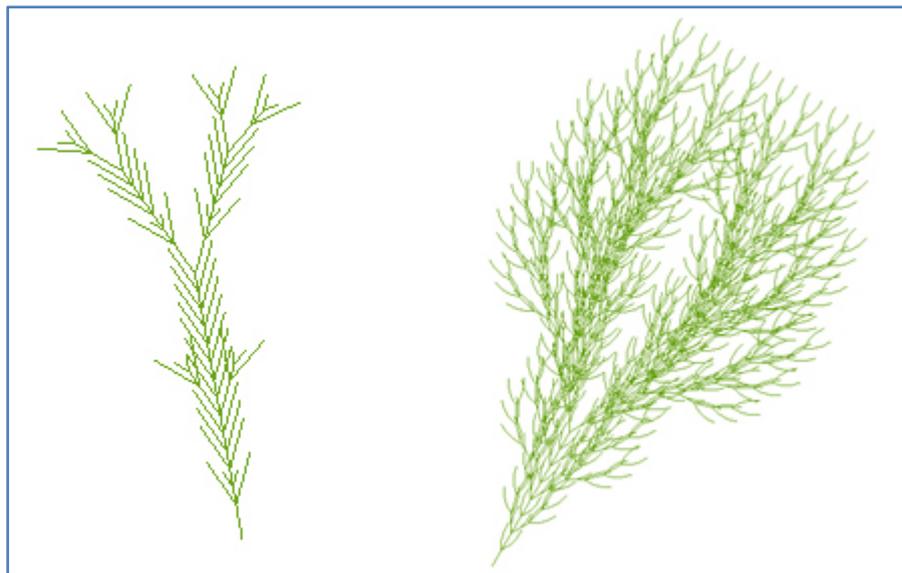


Figura 33: Dos tipos de sistemas L modelados (“Simple plant” y “Nice tree”).



Figura 34: Los árboles son animados y tienen los puntos donde se puede agregar hojas.

Los puntos amarillos de la figura representan los lugares donde se deberían insertar las hojas para darle mayor realismo a la planta generada.

Para terminar la creación de nuevas plantas y entrar en modo de edición, se debe tocar fuera de la zona de edición marcada en la pantalla. En el estado de edición, los contactos del usuario crean círculos de “bloqueo” donde las plantas no pueden crecer y así se puede controlar el crecimiento y la forma de las plantas.

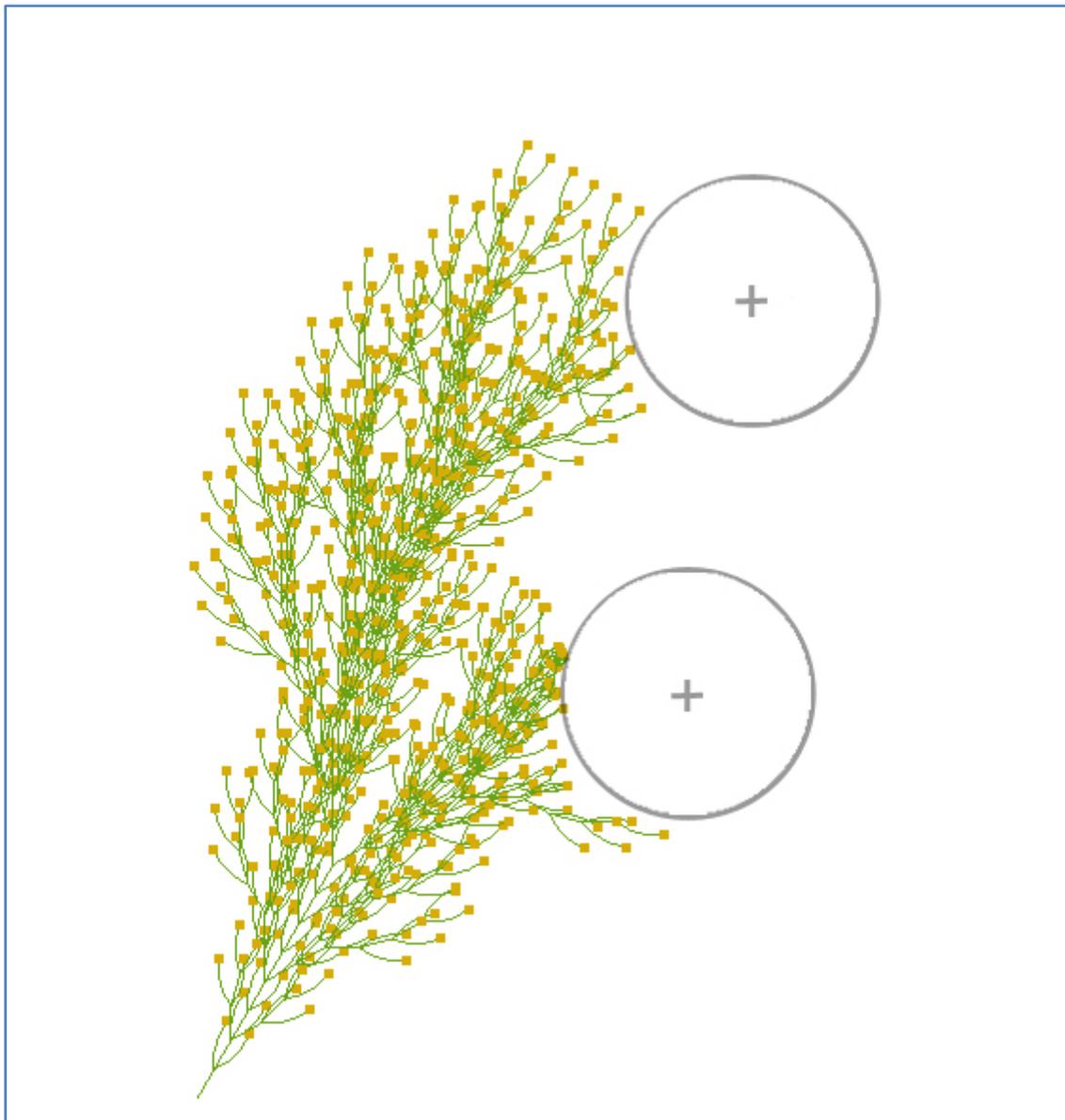


Figura 35: Animación con interacción multitáctil para formar árboles.

Un mismo círculo puede afectar a varias plantas, mientras que las plantas entre sí no se molestan. Se pueden esconder los círculos a la visibilidad del usuario para apreciar el dibujo final.

Para la implementación de la interfaz multitáctil se utilizan las librerías TUIO-CPP [49], para la programación e implementación del cliente, y Tbeta [7] para el servidor, captura y procesamiento de la cámara de infrarrojos.

Tbeta es una aplicación independiente que se encarga de procesar las imágenes capturadas por la cámara de infrarrojos, y calcular la posición de los contactos en un dispositivo virtual. También asigna un identificador único a cada contacto, de manera que se puede llevar una historia de por dónde ha pasado el dedo del usuario por la

pantalla. Tbeta envía los eventos generados a partir del procesamiento de las imágenes mediante una conexión UDP.

Por otra parte, la biblioteca TUIO-CPP facilita la implementación de la parte cliente de aplicaciones multitáctiles. Proporciona una serie de clases C++ que abstraen el acceso a la red y decodifican el protocolo de los paquetes generados por Tbeta. Esta biblioteca también permite la independencia del sistema que procesa las imágenes de la cámara, siempre que respeten el protocolo establecido.

La mayoría de las aplicaciones que usan TUIO-CPP están interesadas únicamente en tres tipos de eventos:

1. detección de un nuevo contacto (cursor, según la terminología de la biblioteca),
2. se ha dejado de detectar un contacto o
3. se ha movido un contacto.

6 Resultados

En esta tesis se han obtenido resultados relacionados con los sistemas multitáctiles y con la interacción en el proceso de generación de modelos procedurales para informática gráfica.

6.1 Sistema Multitáctil

Respecto a los sistemas multitáctiles se ha trabajado en dos sistemas: se ha colaborado en la construcción de un banco de trabajo dentro del marco de un proyecto de la Universidad Politécnica de Valencia y por otra parte se ha diseñado y construido otro sistema portátil de manejo más cómodo y sencillo.

6.1.1 Banco de trabajo multitáctil

Se ha ayudado a completar el banco y se ha conseguido hacer funcionar los demostradores incluidos en sus librerías. El banco de trabajo ya está en funcionamiento y los responsables del proyecto han empezado a implementar algunas aplicaciones multitáctiles adaptadas al mismo.

La superficie de proyección es bastante grande, suficiente para trabajar interactivamente desde el frente. La altura de la mesa es ajustable por lo que se puede acomodar dependiendo de las necesidades. De igual forma, la inclinación de la pantalla es variable entre 0 y 90 grados sobre la horizontal. Eso sí, cambiar la inclinación de la pantalla requiere ajustar el proyector y la cámara de infrarrojos, aunque esto debería ocurrir raras veces.

Uno de los principales problemas del banco de trabajo es que tiene problemas para detectar los contactos de forma fiable en condiciones de mucha luz ambiental. Cuando

esto sucede, se puede hacer funcionar sin la parte FTIR sólo utilizando los blobs producidos por las sombras. Esto conlleva otro problema: las sombras de los dedos o manos varían dependiendo de la iluminación ambiental y del cuarto, además de las sombras proyectadas por los usuarios.

Teniendo la iluminación correcta en cualquiera de los dos casos, el banco de trabajo multitáctil funciona perfectamente. Se detectan perfectamente los blobs y se trabaja fácilmente, sin necesidad de realizar mucha presión sobre la superficie de proyección.

6.1.2 Sistema multitáctil portátil

Durante la realización de esta tesis se diseñó y construyó un sistema multitáctil portátil basado en una pantalla LCD. Las dimensiones de la caja del sistema se calcularon en función del tamaño de la pantalla, el ángulo de visión de la cámara y su distancia mínima para abarcar el espejo. La caja mide 34x53x13 cm para una pantalla LCD de 17" y pesa alrededor de 4 Kg. Utiliza un cable de corriente para el monitor y otro para la iluminación infrarroja y el ventilador, un cable de vídeo VGA para recibir las imágenes y un cable USB para la cámara infrarroja. El coste final fue de alrededor de 150 euros.

El sistema permite controlar el encendido del LCD, el ventilador y la luz infrarroja. Debido a la tecnología LCD y el FTIR, el sistema no puede utilizarse en el exterior, o en habitaciones con mucha iluminación. Es transportable debido a su peso y tamaño, además se puede colocar en cualquier superficie horizontal. El problema del sistema es su fragilidad, debido especialmente al panel LCD y al espejo.

6.2 Generación Procedural

Respecto a la generación de modelos procedurales para informática gráfica se han obtenido los siguientes resultados en esta tesis.

6.2.1 Sistemas de generación procedural actuales

Se ha realizado una evaluación de distintos sistemas de generación procedurales. Estos sistemas son útiles para la creación de muchos objetos que, de otro modo, serían difíciles de crear manualmente. También son útiles cuando deben ser creados en el momento, ahorrando espacio de almacenaje. Los problemas que hemos encontrado son que las herramientas no son sencillas de manejar y que, una vez creados, los modelos son difíciles de editar.

No obstante, existen sistemas que logran un realismo excepcional, sobre todo una vez se ha aprendido a utilizarlos. Algunos permiten crear texturas, calles, árboles, plantas, edificios, puentes, etc. El problema es que todavía se necesitan herramientas para reducir la complejidad de los objetos creados y los recursos necesarios para generarlos.

6.2.2 Sistema de generación procedural multitáctil

Los sistemas de generación procedural existentes en la actualidad tienen en común la complejidad de su uso. Esto viene en gran parte derivado de la cantidad de variables a controlar para obtener el resultado deseado. Como parte del estudio de la utilidad de los interfaces multitáctiles para controlar sistemas complejos, se ha implementado una sencilla aplicación de generación procedural de plantas con un control multitáctil.

El interfaz multitáctil permite controlar de manera natural y en paralelo el proceso de creación y crecimiento procedural. Se ha demostrado cómo controlar el proceso de crecimiento de una planta, para darle forma interactivamente.

7 Conclusiones

El objetivo de esta tesis era la demostración de la viabilidad de los interfaces multitáctiles para el control de aplicaciones complejas.

Se han presentado dos sistemas de interacción multitáctil y una aplicación de generación procedural que aprovecha las características de este tipo de interacción. Los sistemas multitáctiles son de dos tipos distintos: un banco de trabajo y un sistema portátil de bajo coste. Ambos han sido probados y se ha demostrado que funcionan correctamente

El diseño utilizado para construir dichos sistemas permite acercar los dispositivos multitáctiles ópticos al gran público. Este tipo de tecnología permite trabajar con un número arbitrario de puntos de contacto simultáneos, en contraposición a otro tipo de sensores, que limitan dicho número.

Para demostrar la aplicabilidad de estos dispositivos se han estudiado las características de un tipo de herramienta compleja: los sistemas de generación procedural de modelos de plantas para Informática Gráfica. Se ha realizado un estudio de usabilidad de dichos sistemas y se han evaluado sus características principales a nivel de interacción. Una vez estudiados se ha implementado un sistema sencillo que permite construir modelos procedurales utilizando un interfaz multitáctil.

Los resultados obtenidos demuestran que la utilización de interfaces multitáctiles con aplicaciones gráficas complejas facilita y hace más natural su interacción con el usuario. También permite implementar interacciones más complejas, que son muy difíciles de conseguir con el modelo de interacción basado en un único punto de contacto. Como demostrador se ha diseñado una aplicación que permite controlar la forma de la planta interactivamente durante su crecimiento, de forma parecida a como trabaja un escultor.

8 Trabajo Futuro

Durante el uso de los prototipos multitáctiles construidos se han detectado ciertos problemas. El banco de trabajo debe ser un poco más robusto y rígido, para evitar el movimiento al apoyarse o por movimientos bruscos de las manos. Como el proyector en la actualidad no está sujeto al armazón de la mesa, cuando la pantalla tiembla, las imágenes de proyección y de captura se quedan estables, por lo que hay una pequeña pérdida de calidad al tocarla. De igual modo se debe buscar una manera de mejorar la iluminación infrarroja y así evitar los problemas con la iluminación ambiental.

El diseño del banco de trabajo necesita algunas mejoras, tanto en el marco de madera como en la tela usada para cubrir la parte trasera. Por ejemplo, el marco se puede forrar con algún material más cómodo y menos agresivo o pulirse y barnizarse. No obstante el banco actual es un prototipo y se planea crear uno definitivo donde la estructura sea construida por un mecánico profesional.

Con respecto al sistema portátil, los monitores LCD y el FTIR tienen problemas en entornos muy luminosos, por lo que hay que controlar de alguna forma la iluminación de la habitación donde se usa el sistema. El problema se agrava cuando la pantalla mira hacia arriba. En el banco de trabajo este problema se reduce debido a su pantalla orientable.

El siguiente paso es crear un armazón de mayor rigidez, sin aumentar considerablemente el peso. Estamos considerando madera o PVC. También queremos mejorar la portabilidad, añadiendo cierres a la compuerta, asas a la caja y un soporte tanto para el espejo como para las placas de pantalla. Esto evitará su movimiento y aumentará su seguridad.

Al sistema de generación procedural multitáctil queremos agregarle una interfaz con mayor control de edición y navegación, así como el paso a trabajar en 3D. También

queremos usar un motor de sistema L más potente, que permita utilizar varios sistemas L a la vez.

Por último, pensamos ampliar el estudio de la interacción multitáctil teniendo en cuenta no sólo contactos puntuales, sino gestos más complejos. En la actualidad, la interacción multitáctil se basa en un conjunto de *blobs*, donde cada uno viene asociado a un cursor puntual. La utilización de la cámara infrarroja permite obtener la forma detallada del contacto con la mesa. Esto permitiría implementar interacciones más complejas.

Finalmente planea realizar una selección e implementación de una batería de aplicaciones para su adaptación al nuevo paradigma de interacción, y una evaluación de su usabilidad mediante pruebas con usuarios.

9 Publicaciones

El trabajo presentado ha sido en parte publicado en las siguientes dos publicaciones:

1. En [50] se realiza un análisis de las técnicas de generación procedural actuales y, concretamente del software más utilizado orientado a la creación de árboles procedurales. Se estudian las características principales de dicho software, así como su usabilidad. Este trabajo fue publicado en la International Conference on Computational Science (ICCS'09) de la ACM. El artículo fue presentado en Baton Rouge, Louisiana, U.S.A., en Mayo del 2009. Este congreso está considerado dentro del Rank A por la Computing Research and Education Association de Australia (CORE).
2. [50] Este trabajo analiza el hardware multitáctil presentando la pantalla multitáctil portátil de bajo coste. El trabajo describe la pantalla con su volumen y peso reducidos, y sus propiedades de ser portable y de fácil construcción. El trabajo fue enviado al Congreso Español de Informática Gráfica (CEIG'09) patrocinado por Eurographics. El artículo se encuentra aceptado y se presentará en San Sebastián, País Vasco, España, en Septiembre del 2009.

10 Bibliografía

1. Selker T.: Touching the future. *Commun. ACM* 51, 12 (2008).
2. Microsoft: Surface. In <http://www.microsoft.com/surface/>.
3. Rekimoto J.: Organic interaction technologies: from stone to skin. *Commun. ACM* 51, 6 (2008).
4. Valli A., Linari L.: Natural interaction sensitivetable. In CHI '08: CHI '08 extended abstracts on Human factors in computing systems (2008), ACM.
5. Valli A.: Natural interaction. In <http://naturalinteraction.org/>.
6. Han J. Y.: Low-cost multi-touch sensing through frustrated total internal reflection. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology(2005)*, ACM.
7. Natural User Interface: Nui group. In <http://www.nuigroup.com>.
8. Wright A.: Making sense of sensors. *Commun. ACM* 52, 2 (2009).
9. Parish, Y.I.H., Muller, P.: Procedural modeling of cities. In: *SIGGRAPH 2001*, pp. 301–308 (2001)
10. Greuter, S., Parker, J., Stewart, N., Leach, G.: Real-time procedural generation of 'pseudo infinite' cities. In: *GRAPHITE 2003*, pp. 87–94 (2003)
11. Sun, J., Yu, X., Baciú, G., Green, M.: Template-based generation of road networks for virtual city modeling. In: *ACM Symposium on Virtual Reality Software and Technology*, pp. 33–40 (2002)

12. Muller, P., Wonka, P., Haegler, S., Ulmer, A., Gool, L.V.: Procedural modeling of buildings. In: SIGGRAPH 2006, pp. 614–623 (2006)
13. Wonka, P., Wimmer, M., Sillion, F., Ribarsky, W.: Instant architecture. *ACM Transactions on Graphics* 22(3), 669–677 (2003)
14. Martin, J.: Procedural house generation: A method for dynamically generating floor plans. In: *Symposium on interactive 3D Graphics and Games* (2006)
15. Ebert, D., Musgrave, F., Peachey, D., Perlin, K., Worley, S.: *Texturing and Modeling: A Procedural Approach*, 3rd edn. Morgan Kaufmann, San Francisco (2002)
16. Weber, J., Penn, J.: Creation and rendering of realistic trees. In: *SIGGRAPH 1995*, pp. 119–128. ACM, New York (1995)
17. Lluch, J., Camahort, E., Vivó, R.: Procedural multiresolution for plant and tree rendering. In: *AFRIGRAPH 2003* (2003)
18. Roden, T., Parberry, I.: Clouds and stars: efficient real-time procedural sky rendering using 3d hardware. In: *ACE 2005 Int. Conference on Advances in Computer Entertainment Technology*, pp. 434–437 (2005)
19. Prachyabrued, M., Roden, T.E., Benton, R.G.: Procedural generation of stylized 2d maps. In: *ACE 2007: Advances in Computer Entertainment Technology* (2007)
20. Roden, T., Parberry, I.: Procedural Level Generation. In: *Game Programming Gems 5*, pp. 579–588. Charles River Media (2005)
21. Quan, L., Tan, P., Zeng, G., Yuan, L., Wang, J., Kang, S.B.: Image-based plant modeling. In: *SIGGRAPH 2006*, pp. 599–604 (2006)
22. Davidson P.L, Han J. Y.: Synthesis and control on large scale multi-touch sensing displays. In *NIME '06: Proceedings of the 2006 conference on New interfaces for musical expression* (2006), IRCAM — Centre Pompidou.
23. Kaltenbrunner M, Jorda S., Geiger G, Alonso M.: The reactable*: A collaborative musical instrument. In *WETICE '06: Proceedings of the 15th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises* (2006), IEEE Computer Society.
24. Kirton T, Ogawa H., Sommerer C., Mignonneau L.: Pins: a prototype model towards the definition of surface games. In *MM '08: Proceeding of the 16th ACM international conference on Multimedia* (2008), ACM.

25. Han J.: Perceptive pixel. In <http://www.perceptivepixel.com>.
26. De la Rivière J.B., Kervégant C., Orvain E., Dittlo N.: Cubtile: a multi-touch cubic interface. In VRST '08: Proceedings of the 2008 ACM symposium on Virtual reality software and technology (2008), ACM.
27. Echtler F., Klinker G.: A multitouch software architecture. In NordiCHI '08: Proceedings of the 5th Nordic conference on Human-computer interaction (2008), ACM.
28. Moscovich T., Hughes J. F.: Indirect mappings of multi-touch input using one and two hands. In CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems (2008), ACM.
29. Wu M., Balakrishnan R.: Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology (2003), ACM.
30. Hirche J., Bomark P., Bauer M., Solyga P.: Adaptive interface for text input on large scale interactive surfaces. In 3rd. International Workshop on Horizontal Interactive Human Computer Systems (2008), IEEE.
31. Echtler F., Huber M., Klinker G.: Shadow tracking on multi-touch tables. In AVI '08: Proceedings of the working conference on Advanced visual interfaces (2008), ACM.
32. Deix, W.: Real-time rendering of fractal rocks. In: Central European Seminar on Computer Graphics (2003).
33. Prusinkiewicz, P., Hammel, M.: A fractal model of mountains with rivers. In: Graphics Interface 1993 (1993).
34. Lindenmayer, A.: Mathematical models for cellular interaction in development, parts i and ii. *Journal of Theoretical Biology* 18, 280–315 (1968).
35. Kelly, G., McCabe, H.: A survey of procedural techniques for city generation. *ITB Journal* 14 (2006).
36. Prusinkiewicz, P., Hammel, M.S., Mjolsness, E.: Animation of plant development. In: SIGGRAPH 1993, pp. 351–360 (1993).
37. FracTree: <http://archives.math.utk.edu/software/msdos/fractals/fractree>
38. L-System4: <http://www.geocities.com/tperz/L4Home.htm>

39. SpeedTree: <http://www.speedtree.com>
40. Xfrog: <http://www.xfrog.com/>
41. LStudio: <http://algorithmicbotany.org>
42. An Ivy Generator: http://graphics.uni-konstanz.de/~luft/ivy_generator
43. TreeMagik G3: http://www.aliencodec.com/product_treemagik.php
44. Tree[d]: <http://www.frecle.net/forum/viewtopic.php?t=780>
45. Dryad: <http://dryad.stanford.edu>
46. TreeGenerator: <http://www.treegenerator.com>
47. MeshTree Studio: <http://www.ogre3d.org/forums/viewtopic.php?t=25909>
48. Arbaro: <http://arbaro.sourceforge.net>
49. TUIO: <http://www.tuio.org/>
50. de la Re A., Abad F., Camahort E., and Juan M.C.: Tools for Procedural Generation of Plants in Virtual Scenes. In ICCS'09: International Conference on Computational Science (2009). ACM.
51. de la Re A., Abad F., Camahort E., Lluch J.: Pantalla Multitáctil Portátil de bajo coste. In CEIG'09: Congreso Español de Informática Gráfica (2009). IEEE.

11 Anexos

11.1 Glosario

Blobs: Término utilizado para representar las marcas que producen los dedos al tocar una superficie y son reconocidos por el sistema como cursores.

DI: Iluminación Difusa. Es un tipo de arquitectura multitáctil.

Dimensión Hausdorff: Es un número real extendido no negativo, asociado a un espacio métrico. Generaliza la noción de la dimensión de un espacio de vector real. Por ejemplo, un punto sería un cero, una línea el uno, el plano el dos, y así sucesivamente. Existen algunos conjuntos irregulares que tienen una dimensión de Hausdorff no enteros.

DSI: Iluminación Difusa de Superficie. Es un tipo de arquitectura multitáctil.

fps: Cuadros por segundo (por sus siglas en inglés, Frames Per Second).

FTIR: Reflexión Interna Total Frustrada (por sus siglas en inglés, Frustrated Total Internal Reflection). Es un tipo de arquitectura multitáctil.

Generación Procedural: es usado principalmente en la producción de medios. Cuando se utiliza el término, se refiere a que se puede crear contenido “en el aire”, o lo que es lo mismo, al momento de necesitarlo. Dicho contenido puede ser imágenes, geometría, texturas, sonido, música, entre otros). Por lo general está relacionado a aplicaciones de la informática gráfica y diseño de videojuegos.

Gran Angular: Es aquella lente cuya distancia focal es menor a la del objetivo normal, resultando un ángulo de visión mayor al de la visión humana (generalmente entre 60 y 180 grados).

LCD: Pantalla de Cristal Líquido (por sus siglas en inglés, Liquid Cristal Display).

LED: Diodo de Emisión de Luz.

LED-LP: Plano de Luz LED. Es un tipo de arquitectura multitáctil.

Ley de Snell: Fórmula utilizada para calcular el ángulo de refracción de la luz al atravesar dos medios de propagación de luz con índice de refracción distinto.

LLP: Plano de Luz Laser. Es un tipo de arquitectura multitáctil.

LOD: Nivel de detalle (por sus siglas en inglés, Level Of Detal). Técnicas para reducir la complejidad de la representación de un objeto 3D para mejorar la velocidad de la aplicación. Se basa en criterios como la distancia del objeto a la cámara, su importancia o su posición en la pantalla.

Multitáctil: En inglés “multitouch”. Técnica de interacción hombre máquina, así como el hardware que se implementa. Consiste en una pantalla táctil que reconoce simultáneamente múltiples puntos de contacto, así como el software asociado a ésta que permite interpretar dichas interacciones simultáneas.

Plexiglas: tipo de plástico (o acrílico), científicamente llamado Polimetilmetacrilato (PMMA). Llamado así debido a una marca comercial conocida.

TIR: Reflexión Interna Total.

VGA: Adaptador Gráfico de Video.

11.2 Artículos

Tools for Procedural Generation of Plants in Virtual Scenes

Armando de la Re, Francisco Abad, Emilio Camahort, and M.C. Juan

Depto. Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
46022 Valencia, Spain

Abstract. Creating interactive graphics applications that present to the user realistic natural scenes is very difficult. Natural phenomena are very complex and detailed to model, and using traditional modeling techniques takes huge amounts of time and requires skilled artists to obtain good results.

Procedural techniques allow to generate complex objects by defining a set of rules and selecting certain parameters. This allows to speed up the process of content creation and also allows to create objects on-the-fly, when needed. On-demand generation of scenes enables the authors to create potentially infinite worlds.

This survey identifies the main features of the most used systems that implement procedural techniques to model plants and natural phenomena and discuss usability issues.

1 Introduction

Massive multiplayer games model huge environments where lots of players can exchange experiences. To obtain appealing landscapes and interesting game fields, modern games require lots of both geometric and texture assets. This pose a difficult problem since it is very expensive to create lots of different objects, landscapes, characters and so on. It is common to reuse such content in the same game. Changing certain characteristics of the object (i.e., its color, its size) to increase the number of different objects in the game is usually detected by the user, thus reducing the realism of the game.

The resources dedicated to create realistic models could be used to improve game play or include innovative features. Procedural content generation techniques appear to speed up the process of creating content. They are also able to generate content on-the-fly, thus reducing the space requirements.

Recently, automatic content creation systems have been used, for example, to model buildings and cities [1,2], roads [3], buildings [4,5], houses [6], textures [7], vegetation [8,9] and sky [10]. Specifically for games, procedural systems have also been used to model 2D maps [11] and game levels [12].

Usually one of the requirements of games is to present realistic vegetation. This is a difficult goal because natural plants are complex organisms and different factors define its shape and color. It is possible to model a realistic plant

with traditional methods, but it usually results in a huge geometric model, with lots of textures, and it is a time consuming task. Rendering complex models also requires applying some technique of LOD to reduce the actual number of polygons processed in the scene. Some procedural generation algorithms are able to generate multi resolution models [9] and some others are able to generate plants based on images [13].

This work focuses on currently available software applications that use procedural generation algorithms to model plants. We describe the features of the most used applications, and we also study their usability.

The rest of the article is structured as follows. First we talk about previous work on procedural content generation. The following section describes the main features of each surveyed system. We provide a table that compares each aspect of the applications. Conclusions and future work ends the paper.

2 Previous Work

The first procedural techniques were based on recursive functions, and were used to create fractal-like images. A fractal is a fragmented geometry shape, where each fragments is (approximately) a reduced copy of the whole shape (self-similarity). They cannot be classified in the traditional Euclidean geometric system, have a rich structure at arbitrarily small scales and have a Hausdorff dimension greater than its topological dimension. Some generation techniques use fractals to generate plants, rocks [14] and other natural phenomena [15].

Other procedural techniques are based on L-Systems [16]. L-Systems are a variation of formal grammars and are used to simulate the growth of plants, fractals and artificial life. They consist of a set of symbols that can be replaced, an axiom or initial state and a set of production rules. The L-System starts with the axiom that is replaced with the corresponding production rule. Then some parts of the rules are replaced with other rules and so on. The results are interpreted by the renderer as positions, orientations and stack structures. This kind of procedural technique is used to generate complex plants and other natural structures.

Many authors have focused on procedural techniques for city generation [17]. We can find applications to generate cities in a terrain [1], organize and simulate cities with procedural methods [2], create roads and streets [3], green areas, bridges, etc. Other systems model buildings [4], houses [6], facades [5]... These systems have to create different buildings, but have to maintain a common look (for example, to model different buildings built around the same era with the same style). Kelly and McCabe [17] defined seven criteria to evaluate a procedural city generation system: realism, scale, variation of buildings and roads, required input to generate it, efficiency, control to modify the generation, and if it is generated in real time.

Other authors have presented techniques for modeling realistic trees [8], reducing the polygon count of the plant model using procedural techniques [9], image-based modeling of plants [13] or animating plants [18].

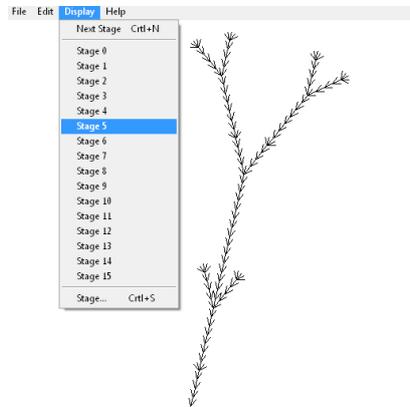


Fig. 1. FractTree allows to render separately each stage of the derivation



Fig. 2. The preview and navigation screen of L-System4

3 Applications

3.1 FractTree [19]

This application is one of the precursors in the generation of fractal plants. It creates only 2D models, and uses L-Systems and a step by step generation with detail level for derivation rules shown in Figure 1. The application creates the plant replacing the symbols in the derivation with drawing primitives. It is a very simple program but it can be used to understand the basics of L-Systems.

3.2 L-System4 [20]

It is also based on L-Systems, and generates detailed 3D plants and objects (see Figure 2). The navigation is somewhat restricted but it is enough to examine the object. One problem of this application is that the user has to know how L-Systems work to create or change one.

3.3 LStudio [21]

LStudio provides several tools to create realistic plants as shown in Figure 3.a. It is based on a modified bracketed L-System to generate trunks, branches and the position of leaves, flowers and petals. These terminals are modeled in the interactive vector editor shown in Figure 3.b. This system is suitable to generate small plants like flowers, grass and bushes, rather than trees.

3.4 An Ivy Generator [22]

It is a generator of Ivy plants that allows the user to decide where to grow them on an imported 3D scene by defining a seed. It has simple tools to change the appearance of the plants but the results are very realistic. It takes into account the gravity and the capacity of the plant to grow to create climbing or hanging plants (Figure 4).

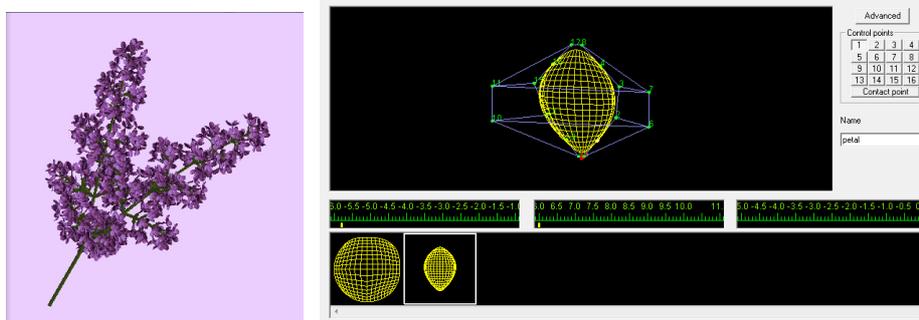


Fig. 3. a) Plant created with LStudio, and b) its editor for modeling leaves, flowers and petals

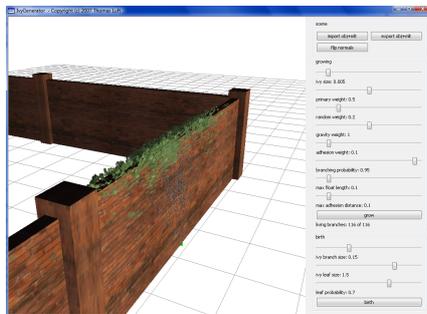


Fig. 4. An Ivy Generator example

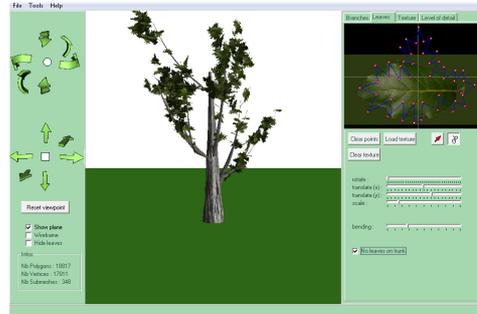


Fig. 5. Treegenerator GUI

3.5 TreeGenerator [23]

This application has a control panel to control the tree generation. Figure 5 shows the leaf editor. The resulting leaves look real when isolated, but groups of leaves do not look realistic. One of the causes is that the program has a limited number of recursion levels to generate branches and leaves. The tools to modify the tree and create different instances are also limited.

3.6 TreeMagik G3 [24]

This tool provides a trunks, branches and roots generator. The foliage is provided by the program and it is rendered as a set of billboards. It generates very good results as shown in Figure 6. It is also able to generate a billboard of the entire tree. It provide textures for the trunk and leaves, and the user can add textures.

3.7 MeshTree Studio [25]

It produces very realistic trees but the user interface is not very user friendly and there is no help or tutorial. Once the initial learning curve is overcome,

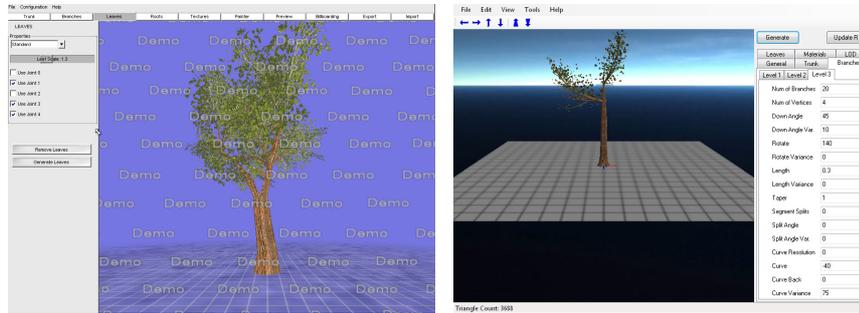


Fig. 6. Example of a tree generated by TreeMagikG3

Fig. 7. Example generated with Meshtree Studio

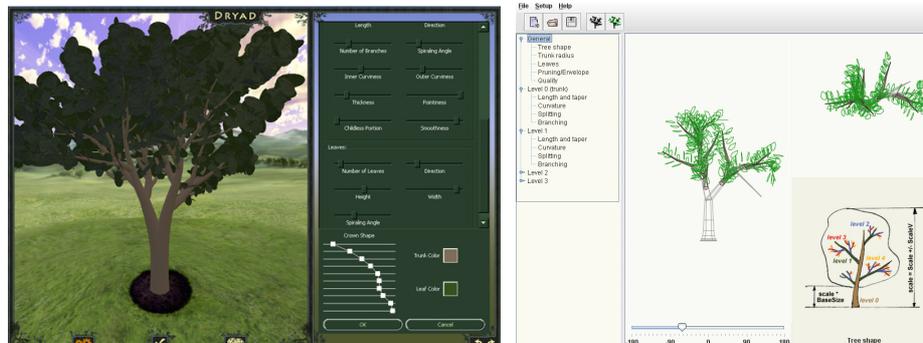


Fig. 8. Dryad generates trees quickly, but they are not very realistic

Fig. 9. The help system of Arbaro provides information about the parameters to control the generation

the user can generate very appealing trees with a low number of polygons. One restriction is that only generate *.mesh* files.

3.8 Dryad [26]

This is a freeware tree generator, but it is not open source. It provides an online gallery of trees that looks like a forest, where the user can select a tree and change its parameters. The properties of two different trees can be combined to create a new tree. The trees created by the users can be planted in the online gallery and shared with other users. A disadvantage of this system is that it only generates high resolution trees and they are not very realistic (Figure 8).

3.9 Arbaro [27]

This is the only Java-based system evaluated and generates good results. It is well documented, but its interface is not very friendly. It has some errors exporting trees. Figure 9 shows the flexibility to control parameters like number of

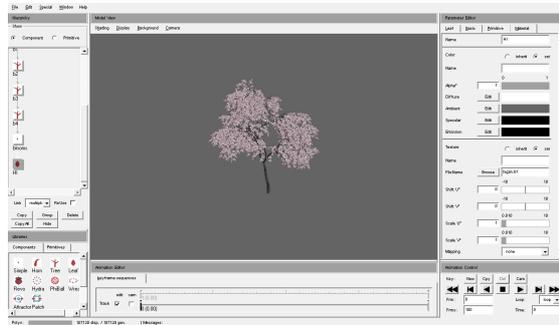


Fig. 10. Xfrog Graphical User Interface

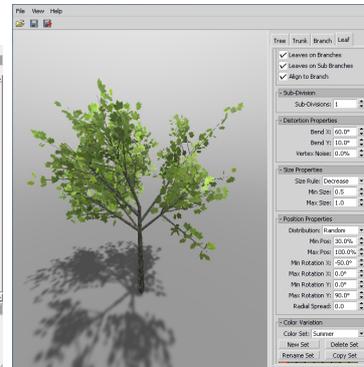


Fig. 11. Tree[d] produces good results but it is difficult to control variations

levels, radius, division and curvatures of branches and trunk. It provides overlay graphics aids to help the user to understand each parameter.

3.10 SpeedTree [28]

It's one of the most used and renowned products in the creation of videogames that present natural scenes. It provides a powerful, complete, and efficient renderer, a modeler and real time generator. The scenes created with this engine are very realistic, and it has also been designed to be used in interactive systems. This is the most expensive system described in this survey, and we could not test it.

3.11 Xfrog [29]

It is similar to SpeedTree and it is also used in the videogame industry. It can generate very realistic scenes, with different objects and a wide variety of plants. It is more affordable than SpeedTree. Its learning curve is also very steep, and takes time to obtain a convincing tree.

3.12 Tree[d] [30]

It is very easy to create random trees and generates very realistic examples. It is difficult to modify or start a new one, and there is not many different type of trees.

4 Conclusions and Future Work

The purpose of this work was to study the current procedural generation systems to model natural plants. We have classified these tools into several profiles, depending on the features of the systems and the requirements of the user: FractTree and L-System4 is suitable for students, since they help to learn and

understand L-Systems. Speedtree and Xfrog are suitable for game and movie companies because they generate the most realistic trees, but at the expense of a steep learning curve and a high cost. L-Studio also creates realistic flower plants but it requires a lot of learning time. An Ivy Generator makes a scene or object look abandoned or alive because of the added Ivy plant, and because it is free, it can be used by everyone. Treemagik G3 and Tree[d] also generate good results and are cheap or free.

Some systems can be used by 3D cartoon animators to generate trees. Dryad, TreeGenerator and L-System4 can generate cartooned trees instead of realistic plants. For the casual user FractTree, dryad, MeshTree studio, TreeGenerator and An Ivy generator provide a lot of parameters to select and different results.

Our group is currently developing a framework for procedural modeling of synthetic models. This tool will allow the user to select different techniques to generate geometry and textures inside the same environment.

References

1. Parish, Y.I.H., Muller, P.: Procedural modeling of cities. In: SIGGRAPH 2001, pp. 301–308 (2001)
2. Greuter, S., Parker, J., Stewart, N., Leach, G.: Real-time procedural generation of ‘pseudo infinite’ cities. In: GRAPHITE 2003, pp. 87–94 (2003)
3. Sun, J., Yu, X., Baciú, G., Green, M.: Template-based generation of road networks for virtual city modeling. In: ACM Symposium on Virtual Reality Software and Technology, pp. 33–40 (2002)
4. Muller, P., Wonka, P., Haegler, S., Ulmer, A., Gool, L.V.: Procedural modeling of buildings. In: SIGGRAPH 2006, pp. 614–623 (2006)
5. Wonka, P., Wimmer, M., Sillion, F., Ribarsky, W.: Instant architecture. *ACM Transactions on Graphics* 22(3), 669–677 (2003)
6. Martin, J.: Procedural house generation: A method for dynamically generating floor plans. In: Symposium on interactive 3D Graphics and Games (2006)
7. Ebert, D., Musgrave, F., Peachey, D., Perlin, K., Worley, S.: *Texturing and Modeling: A Procedural Approach*, 3rd edn. Morgan Kaufmann, San Francisco (2002)
8. Weber, J., Penn, J.: Creation and rendering of realistic trees. In: SIGGRAPH 1995, pp. 119–128. ACM, New York (1995)
9. Lluch, J., Camahort, E., Vivó, R.: Procedural multiresolution for plant and tree rendering. In: AFRIGRAPH 2003 (2003)
10. Roden, T., Parberry, I.: Clouds and stars: efficient real-time procedural sky rendering using 3d hardware. In: ACE 2005 Int. Conference on Advances in Computer Entertainment Technology, pp. 434–437 (2005)
11. Prachyabrued, M., Roden, T.E., Benton, R.G.: Procedural generation of stylized 2d maps. In: ACE 2007: Advances in Computer Entertainment Technology (2007)
12. Roden, T., Parberry, I.: Procedural Level Generation. In: *Game Programming Gems 5*, pp. 579–588. Charles River Media (2005)
13. Quan, L., Tan, P., Zeng, G., Yuan, L., Wang, J., Kang, S.B.: Image-based plant modeling. In: SIGGRAPH 2006, pp. 599–604 (2006)
14. Deix, W.: Real-time rendering of fractal rocks. In: Central European Seminar on Computer Graphics (2003)

15. Prusinkiewicz, P., Hammel, M.: A fractal model of mountains with rivers. In: Graphics Interface 1993 (1993)
16. Lindenmayer, A.: Mathematical models for cellular interaction in development, parts i and ii. *Journal of Theoretical Biology* 18, 280–315 (1968)
17. Kelly, G., McCabe, H.: A survey of procedural techniques for city generation. *ITB Journal* 14 (2006)
18. Prusinkiewicz, P., Hammel, M.S., Mjolsness, E.: Animation of plant development. In: SIGGRAPH 1993, pp. 351–360 (1993)
19. FracTree: <http://archives.math.utk.edu/software/msdos/fractals/fractree>
20. L-System4: <http://www.geocities.com/tperz/L4Home.htm>
21. LStudio: <http://algorithmicbotany.org>
22. An Ivy Generator: http://graphics.uni-konstanz.de/~luft/ivy_generator
23. TreeGenerator: <http://www.treegenerator.com>
24. TreeMagik G3: http://www.aliencodec.com/product_treemagik.php
25. MeshTree Studio: <http://www.ogre3d.org/forums/viewtopic.php?t=25909>
26. Dryad: <http://dryad.stanford.edu>
27. Arbaro: <http://arbaro.sourceforge.net>
28. SpeedTree: <http://www.speedtree.com>
29. Xfrog: <http://www.xfrog.com/>
30. Tree[d]: <http://www.frecle.net/forum/viewtopic.php?t=780>

Summary

This section includes a table comparing the main features of each system. The features studied are:

- 2D and 3D: Capacity to generate 2D and/or 3D objects.
- IDE: Integrated editor to create the plants.
- Definition Language: Input for plant definition. It can be LSys [] (bracketed L-Systems) or GUI (Graphical User Interface).
- Navigation: Navigation in the rendered scene (Zoom, move, rotation).
- Geometry Generation: It can generate geometry.
- Import, Export: file formats accepted by the application.
- Released, Updated: First release and last update dates.
- Derivation Control: Control of the derivation of plants.
- Multiple Objects: Render multiple objects at the same time.
- Object types: Objects that the system can work with.
- Textures: Capacity to use textures in plant creation.
- Purpose: Main purpose of the application.
- Usability: Easy of use in a 1-10 scale (1 difficult, 10 easy).
- Documentation: Quality of the documentation in a 0-10 scale (0 no documentation, 1 poor, 10 complete and user friendly).
- Debugging: Tools for debugging the generation process.
- Modeling Speed: Time to create a target plant in each system. It could be from a blank project or modifying an existing example. The plant has these requirements: An initial branch (or trunk) and a separation in two branches, five derivations and medium-sized round leaves.

Table 1. Main characteristics of the surveyed applications. Legend for “Modeling Speed” row: : 1, fast using examples and knowing L-Systems; 2, very slow, required modifying existing examples; 3, very fast, requires an external object; 4, fast to build but difficult to modify; 5, very fast; 6, fast but export does not work; 7, normal and 8, we could not test it

	FracTree	L-System4	L-Studio Botany Alg	An Ivy Generator	TreeGenerator	TreeMagik G3
2D	x	-	x	-	-	-
3D	-	x	x	x	x	x
IDE	x	x	x	x	x	x
Definition Language	Lsys □	Lsys □	Mod. Lsys □	GUI	GUI	GUI
Navigation	-	x	x	x	x	x
Geometry Generation	-	x	x	x	x	x
Image Generation	-	x	x	x	x	x
Import	-	dxif	-	obj	-	b3d
Export	bmp	dxif, bmp, jpeg	rgb, tga, bmp	obj	obj, 3ds, dxf	several (obj, wrf)
License	Shareware	Freeware	Freeware	Freeware	1.3 & 2.0	Demo
Price	10 EUR	0	0	0	\$0 & \$49	\$49.95
Released	1993	2000	1999	2007	2006	2006
Updated	No	2004	2004	2008	No	No
Derivation control	x	x	-	x	x	x
Multiple objects	-	x	x	x	-	-
Object types	Lsystems	dxif, Lsystems	Plants Fractals	3D Objects	Trees	Trees
Textures	-	x	-	x	x	x
Purpose	Fractal trees	Lsystems	Plants	Ivy Plants	Trees	Trees
Usability	6	8	8	9	8	7
Documentation	6	2	10	4	6	0
Debugging	Level control	-	-	Level control	Instant update	Level control
Modeling Speed	1	1	2	3	4	5

Table 1. (continued)

	MeshTree	Studio	Dryad	Arbaro	XFrog	SpeedTree	Tree[d]
2D	-	-	-	-	-	-	-
3D	x	x	x	x	x	x	x
IDE	x	x	x	x	x	x	x
Definition Language	GUI	GUI	GUI	GUI	GUI	GUI	GUI
Navigation	x	x	x	x	x	x	x
Geometry Generation	x	x	x	x	x	x	x
Image Generation	-	-	-	-	-	-	-
Import	-	-	-	xml	-	-	-
Export	.mesh	obj	obj	obj, povray, dxf	png, jpg	several	x, b3d
License	Freeware	Freeware	Freeware	Freeware	Trial, lite, full	Trial, full	Freeware
Price	0	0	0	0	\$300/\$400	\$8495,	0
Released	2007	2007	2007	2003	1996	2002	2002
Updated	No	2008	2008	2004	2002	2009	2008
Derivation control	x	-	-	x	x	x	-
Multiple objects	-	-	-	-	-	-	-
Object types	Trees	Trees	Trees	xml	Plants	Plants	Trees
Textures	x	-	-	-	x	x	x
Purpose	Trees	Trees	Trees	Trees	Plants	Plants	Trees
Usability	6	5	5	5	9		5
Documentation	0	0	0	6	10		0
Debugging	Level control	Instant update	Instant update	Instant update	a lot		Instant update
Modeling Speed	7	5	5	6	2	8	5

Pantalla multitáctil portátil de bajo coste

A. de la Re¹, F. Abad², E. Camahort² and J. Lluch²

¹Departamento de Sistemas Informáticos y Computación

²Instituto Universitario de Automática e Informática Industrial

Universidad Politécnica de Valencia

Camino de Vera S/N 46022 Valencia, España

Abstract

La tecnología multitáctil ha comenzado a introducirse en nuestra vida cotidiana, impulsada principalmente por los smartphones. Algunas compañías han lanzado al mercado pantallas de ordenador multitáctiles, presentando formas alternativas de interacción.

Los principales problemas de esta tecnología emergente son el alto coste y el tamaño de los dispositivos. Dichos problemas la restringen habitualmente a ferias, empresas o laboratorios de investigación.

Recientemente, grupos de entusiastas e investigadores han publicado algunas técnicas, manuales y vídeos de cómo construir pantallas multitáctiles. El problema es que la mayoría de ellas están basadas en un proyector, por lo que normalmente ocupan demasiado espacio.

En este trabajo presentamos una pantalla multitáctil portátil de bajo coste. Tiene tanto un volumen como un peso reducidos y es fácil de construir.

Palabras clave: Multitáctil, Portátil, Pantalla LCD

Categories and Subject Descriptors (according to ACM CCS):

B.4.2 [Hardware]: Input/Output Display—Image Display

H.5.2 [Information Interfaces and Presentation]: User Interfaces—Input devices and strategies

I.3.1 [Computer Graphics]: Hardware Architecture—Input devices

1. Introducción

Uno de los objetivos de la industria informática es desarrollar tecnologías amigables, innovadoras y atractivas para el usuario, que faciliten su uso y aprendizaje. La mayoría de periféricos tradicionales de entrada y salida (teclado, ratón, trackball, cámara web, monitor, proyector, etc.) han evolucionado sorprendentemente poco en los últimos años. La interacción basada en un dispositivo apuntador, un teclado y un monitor ha variado poco en sus más de 35 años de edad. Una de las tendencias tecnológicas más recientes, que ha despertado interés en la comunidad investigadora y ha empezado a introducirse en el mercado de consumo, son las interfaces multitáctiles. Dichas interfaces permiten atender a más de un punto de contacto a la vez, y funcionan como un periférico de entrada y salida, sustituyendo a los tres dispositivos prin-

cipales que componen un ordenador moderno: el monitor, el ratón y el teclado.

Las interfaces multitáctiles se pueden encontrar hoy en día en el mercado de telefonía móvil [Sel08], las pantallas de ordenador [Mic], bancos de trabajo [Rek08, VL08] e incluso en pizarras o pantallas de proyección [Val]. Sin embargo, siguen siendo caras y difíciles de construir. Recientemente han surgido propuestas innovadoras para implementar interfaces multitáctiles con componentes comunes [Han05] así como software y librerías para facilitar su programación [Nat].

Los componentes principales de las pantallas multitáctiles actuales son un proyector, una cámara web y una superficie de proyección. Por lo general este tipo de pantallas necesitan bastante espacio para situar todos los componentes. Para mostrar una imagen, el proyector necesita una distancia mínima de proyección. Dicho proyector, además de ser costoso,

frágil, y por lo general pesado, necesita un mantenimiento regular, debido a lo reducido de la vida útil de su fuente de luz.

Hay diferentes tipos de tecnologías para la captura de la información táctil [Sel08, Wri09], siendo las cámaras infrarrojas la tecnología más usada, más fácil de implementar y de menor coste. Sin embargo, unas de las principales limitaciones de estas cámaras son su ángulo de visión y la necesidad de situarlas a una distancia mínima de la superficie de proyección.

Teniendo en cuenta todas estas limitaciones, hemos construido un sistema que no se basa en un proyector, sino en una pantalla LCD. Usando una cámara web con una lente gran angular y un espejo, hemos conseguido un sistema multitáctil portátil de bajo coste.

2. Trabajo previo

Se pueden encontrar diversas aplicaciones en las que se han utilizado interfaces multitáctiles, como sintetizadores de música [DH06, KJGA06], juegos de mesa [KOSM08], difusión de noticias [Han], paredes de escalada interactivas [Val] o el manejo de aplicaciones 3D [DIRKOD08].

La arquitectura multitáctil está formada normalmente por cuatro componentes: la pantalla, el proyector, la distribución infrarroja y la cámara infrarroja. Existen distintas formas de combinarlos, tal y como se verá a continuación.

Las cinco tecnologías para la creación de sistemas multitáctiles más estudiadas en la actualidad son: Frustración de reflexión total interna (FTIR) [Han05], Iluminación Difusa (DI) [Mic], Plano de Luces Laser (LLP), en investigación por Microsoft, Iluminación Difusa de Superficie (DSI) y Plano de Luz LED (LED-LP). Las tres últimas se pueden encontrar en [Nat]. También hay otros tipos de sensores para construir interfaces multitáctiles. Algunos de estos sensores capturan proximidad, acústica, capacitancia, resistividad, movimiento, orientación y presión.

Reflexión Interna Total Frustrada (FTIR). La Reflexión Interna Total (TIR) es un fenómeno óptico que se produce cuando un rayo intenta pasar de un material a otro con menor índice refractivo con un ángulo de incidencia mayor que el ángulo crítico respecto a la normal de la superficie, por lo que el rayo se refleja internamente. Utilizando este principio en un acrílico con leds infrarrojos a sus costados, se produce reflexión interna de luz infrarroja. Al tocar el usuario su superficie, se dice que la reflexión se frustra, permitiendo que los rayos de infrarrojo se escapen del acrílico, y sean capturados por la cámara.

Iluminación Difusa (DI). Este tipo de arquitectura sólo necesita una superficie de proyección que deje pasar luz infrarroja, por lo general un acrílico blanco. Es bastante complejo realizar una distribución uniforme de la luz infrarroja en este tipo de arquitectura, por lo que a menudo resulta que ciertas

partes de la pantalla reciben más luz que otras. Dependiendo de la posición relativa entre la luz y la pantalla, existen dos tipos de iluminación difusa: frontal y trasera. Al tocar el usuario la pantalla, la iluminación frontal provoca sombras sobre ella. La iluminación trasera, por otro lado, provoca que el contacto refleje la luz. La iluminación trasera necesita algún tipo de difusor para distribuir la luz, y frecuentemente este difusor funciona también como superficie de proyección. Estas técnicas tienen como ventaja poder reconocer figuras en el punto de contacto (y no simplemente un *blob*), pero como desventaja no funcionan bien en lugares iluminados.

Plano de Luz Laser (LLP). Esta tecnología necesita una superficie de proyección que deje pasar luz infrarroja. Se dispone una malla de luz láser que abarca toda la superficie por el lado donde se encuentra el usuario. Al romper el usuario el plano, hay luz que es rebotada hacia abajo y una cámara detecta la luz infrarroja producida. El gran problema de utilizar luz láser es la seguridad del usuario, por lo que es necesario protección especial para los ojos.

Iluminación Difusa de Superficie (DSI). Similar a la arquitectura FTIR, utiliza los leds infrarrojos al costado de un acrílico especial, pero su funcionamiento es como un DI. Dicho acrílico tiene partículas dentro del material que actúan como miles de pequeños espejos. Al irradiar con luz infrarroja en el borde del acrílico, es distribuida y lanzada en las dos direcciones.

Parejo al desarrollo hardware, han aparecido diferentes sistemas software en forma de aplicaciones finales o bibliotecas. Típicamente dicho software se desarrolla *ad hoc* para cada plataforma. Ehtler y Klinker [EK08] proponen una arquitectura que intenta crear una aplicación base o estándar, unificando las similitudes y características del software existente.

Para reducir el espacio necesario para construir un dispositivo multitáctiles, hay grupos que proponen utilizar varias cámaras [Nat]. El problema de dicha solución, aparte del aumento del coste del sistema es también el aumento del procesamiento a realizar en cada *frame*.

3. Diseño del sistema

Con el objetivo de construir una pantalla multitáctil orientada a su uso en casa, se plantearon los siguientes requisitos:

- ocupar poco espacio,
- ser portátil, por lo que además, debía pesar poco
- ser barata.

Se descartó el uso de un proyector debido a dos limitaciones: son costosos y necesitan bastante espacio de proyección, incluso utilizando espejos. En vez de usar un proyector, se usó un monitor LCD, que elimina por completo la necesidad del espacio de proyección. Un monitor plano está compuesto básicamente por una pantalla semitransparente donde

se crean las imágenes y por una fuente de iluminación trasera. La pantalla se basa en la tecnología LCD, que permite controlar píxel a píxel la cantidad de luz que pasa a través, y por lo tanto, el color es percibido por el usuario.

Sin embargo, uno de los inconvenientes de las pantallas LCD es su fragilidad, lo que nos llevó a extremar el cuidado en el diseño del chasis del dispositivo para evitar su rotura durante su transporte. La figura 1 muestra un esquema con los elementos principales de nuestro sistema.

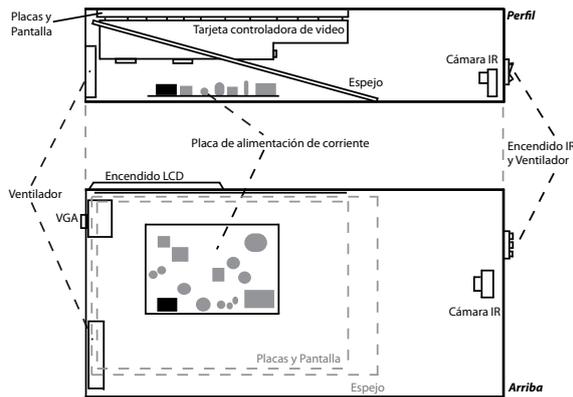


Figure 1: Diseño y distribución de los componentes en la caja

El ángulo de visión de las cámaras impone una distancia mínima para abarcar el objetivo. Sin embargo, recientemente han aparecido webcams con lentes de gran angular que reducen la distancia necesaria para cubrir completamente la pantalla de proyección. En nuestro sistema se usó una cámara de 80 grados de ángulo de visión en horizontal y 60 fps, modelo PHILIPS SPC1001NC/00. Para permitir que la cámara capture la luz infrarroja, se le quitó el filtro de infrarrojos que incorpora la mayoría de cámaras de consumo y se le instaló un filtro paso alto de 850nm.

Se han utilizado leds infrarrojos OSRAM SFH485, que emiten luz a 880nm. Una de las opciones que se barajaron al inicio del proyecto fue implementar la técnica de DI, aprovechando el espejo, pero debido a los problemas encontrados se optó por usar FTIR. Para ello, se instaló un acrílico transparente de 3mm encima de la pantalla LCD, sin dejar espacio, iluminado en dos lados por 12 leds.

Los dispositivos multitáctiles de los que se ha tenido conocimiento y que utilizan la tecnología LCD toman normalmente la forma de una mesa, con la iluminación situada en el suelo, paralela a la pantalla. El resto de componentes se distribuyen para bloquear la menor cantidad de luz posible. El problema de este diseño es que típicamente necesita una altura considerable.

Nuestro diseño, por otra parte, usa un espejo para reducir la altura del dispositivo, lo que introduce un problema a la hora de iluminar la pantalla. El diseño reutiliza la iluminación original del LCD. Dicha fuente de iluminación consiste en un acrílico transparente iluminado por tubos fluorescentes a sus costados, una lámina de plástico opaco en la parte trasera y tres láminas de distintos plásticos enfrente. Utilizando la técnica de TIR, una de las caras del acrílico cuenta con pequeños puntos marcados que rompen el efecto y lanzan la luz hacia el plástico opaco, en la parte trasera del monitor. Las láminas delanteras son dos hojas difusoras y una hoja bloqueadora de infrarrojos. Se retiraron la lámina bloqueadora de infrarrojos y el plástico opaco de la parte trasera, y se invirtió el sentido del acrílico. Los puntos marcados se orientaron hacia el LCD para dejar salir la luz, pero las hojas difusoras no fueron suficientes para evitar que se distinguieran los puntos en el LCD. Retirando las hojas difusoras y usando un acrílico blanco se consiguió una iluminación uniforme. La figura 2 muestra la estructura de la pantalla.

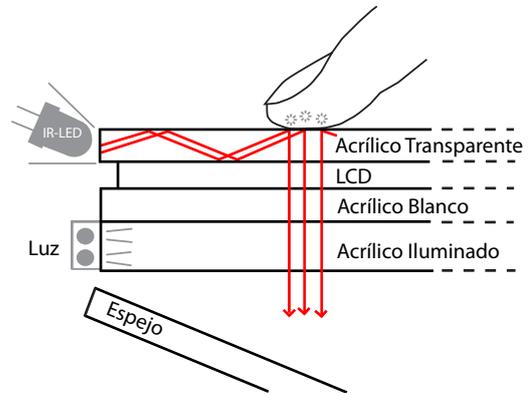


Figure 2: Estructura de las capas que forman la pantalla.

Un ventilador en la caja evita el aumento de temperatura en el interior debido a la pantalla y su alimentación, la iluminación y las resistencias de los infrarrojos. También se han instalado dos interruptores para encender la iluminación infrarroja y el ventilador.

4. Resultados

Las dimensiones de la caja se calcularon en función del tamaño de la pantalla, el ángulo de visión de la cámara y su distancia mínima para abarcar el espejo. La caja mide $34 \times 53 \times 13$ cm para una pantalla LCD de 17" y pesa alrededor de 4 Kg. Utiliza un cable de corriente para el monitor y otro para la iluminación infrarroja y el ventilador, un cable de vídeo VGA para recibir las imágenes y un cable USB para la cámara infrarroja (véase la figura 3). El coste final fue de alrededor de 150 euros.

Debido a la tecnología LCD y el FTIR, el sistema no puede utilizarse en el exterior, o en habitaciones con mucha ilu-



Figure 3: Detalles de la pantalla

minación. Es transportable debido a su peso y tamaño, además se puede colocar en cualquier superficie horizontal. El problema del sistema es su fragilidad, especialmente debido al panel LCD y al espejo.

5. Conclusiones y Trabajo futuro

Los monitores LCD y el FTIR tienen problemas en entornos muy luminosos, por lo que hay que controlar de alguna forma la iluminación de la habitación donde se use el sistema. El problema se agrava debido a que la pantalla mira hacia arriba (véase la figura 4).



Figure 4: Pantalla multitáctil funcionando

Sin embargo, este sistema permite acercar los dispositivos

multitáctiles ópticos al gran público. Este tipo de tecnología permite trabajar con un número arbitrario de puntos de contactos simultáneos, en contraposición a otro tipo de sensores, que limitan dicho número.

Como trabajo futuro se planea crear un armazón de mayor rigidez, sin aumentar considerablemente el peso, ya sea de madera o pvc. También se piensa en mejorar la portabilidad, añadiendo cierres y asas.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Programa de Apoyo a la Investigación y Desarrollo (PAID-06-08) de la Universidad Politécnica de Valencia.

References

- [DH06] DAVIDSON P. L., HAN J. Y.: Synthesis and control on large scale multi-touch sensing displays. In *NIME '06: Proceedings of the 2006 conference on New interfaces for musical expression* (2006), IRCAM — Centre Pompidou. 2
- [dlRKOD08] DE LA RIVIÈRE J.-B., KERVÉGANT C., ORVAIN E., DITTLON N.: Cubtile: a multi-touch cubic interface. In *VRST '08: Proceedings of the 2008 ACM symposium on Virtual reality software and technology* (2008), ACM. 2
- [EK08] ECHTLER F., KLINKER G.: A multitouch software architecture. In *NordiCHI '08: Proceedings of the 5th Nordic conference on Human-computer interaction* (2008), ACM. 2
- [Han] HAN J.: Perceptive pixel. In <http://www.perceptivepixel.com>. 2
- [Han05] HAN J. Y.: Low-cost multi-touch sensing through frustrated total internal reflection. In *UIST '05: Proceedings of the 18th annual ACM symposium on User interface software and technology* (2005), ACM. 1, 2
- [KJGA06] KALTENBRUNNER M., JORDA S., GEIGER G., ALONSO M.: The reactable*: A collaborative musical instrument. In *WETICE '06: Proceedings of the 15th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises* (2006), IEEE Computer Society. 2
- [KOSM08] KIRTON T., OGAWA H., SOMMERER C., MIGNONNEAU L.: Pins: a prototype model towards the definition of surface games. In *MM '08: Proceeding of the 16th ACM international conference on Multimedia* (2008), ACM. 2
- [Mic] MICROSOFT: Surface. In <http://www.microsoft.com/surface/>. 1, 2
- [Nat] NATURAL USER INTERFACE: Nui group. In <http://www.nuigroup.com>. 1, 2
- [Rek08] REKIMOTO J.: Organic interaction technologies: from stone to skin. *Commun. ACM* 51, 6 (2008). 1
- [Sel08] SELKER T.: Touching the future. *Commun. ACM* 51, 12 (2008). 1, 2
- [Val] VALLI A.: Natural interaction. In <http://naturalinteraction.org/>. 1, 2
- [VL08] VALLI A., LINARI L.: Natural interaction sensitivetable. In *CHI '08: CHI '08 extended abstracts on Human factors in computing systems* (2008), ACM. 1
- [Wri09] WRIGHT A.: Making sense of sensors. *Commun. ACM* 52, 2 (2009). 2