Proyecto mOpP

Desarrollo de una herramienta para royecciones de video sobre un volúmen.

El presente documento es una descripción del trabajo de investigación llevado a cabo en el EmmeLab[7] durante los años 2011 y 2012.

Director: Ing. Emiliano Causa
Autores: Francisco Alvarez Lojo, Ezequiel Rivero, Ariel Uzal.
(integrantes del Laboratorio EmmeLab)

Resumen

El video-mapping, también conocido como projection-mapping o sencillamente mapping, consiste en la proyección de imágenes sobre una superficie con más de un plano y/o con una posición que no es perpendicular al haz central del cono de luz. El problema principal que esto genera es que la imagen proyectada se verá deformada pues la misma recorrerá diferentes distancias antes de llegar a la superficie de proyección. En los últimos años, este recurso ha sido utilizado cada vez con mayor frecuencia. Gracias a la mayor accesibilidad de equipos tecnológicos avanzados y a la existencia en internet de una mayor cantidad de comunidades y foros dedicándose a este tema. Sin embargo, sólo existen unas pocas herramientas desarrolladas específicamente para trabajar con mapping que tengan en cuenta la mayoría de los problemas que pueden ser encontrados durante el desarrollo de una pieza de este tipo.

Es por eso que el grupo Emmelab comenzó a desarrollar en el 2011 un trabajo que utilizaría mapping y un software capaz de abordar sus principales problemas.

El resultado del proyecto se denominó mOpP, y consta de una escultura envuelta completamente por una proyección de imágenes interactivas generadas en tiempo real.

mapping

Projection mapping

proyecciones volumétricas

realidad aumentada

processing

Introducción al video-mapping.

En estos años hemos visto aparecer, cada vez con más frecuencia, el uso de un recurso visual llamado video-mapping. Este recurso es una manifestación de realidad aumentada, donde la intención principal es generar una fusión verosímil entre la imagen proyectada y el aspecto de un objeto físico, ya sea una escultura, una escenografía o la fachada de un edificio.

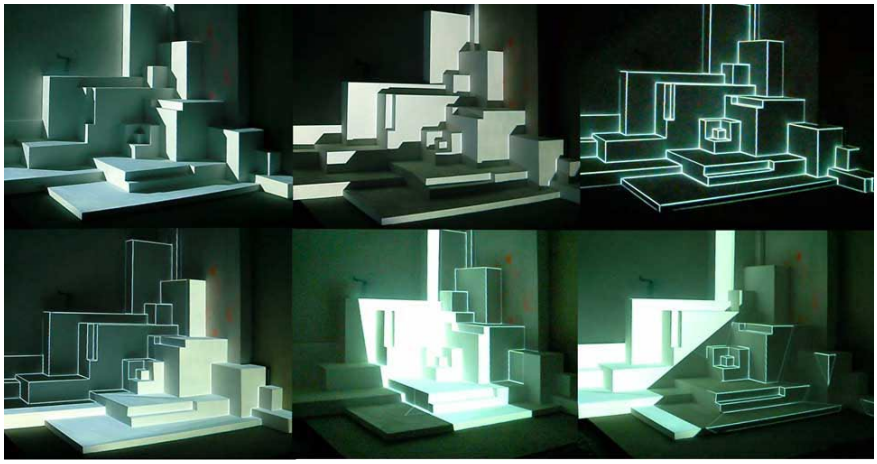


Figura 1

"Augmented Sculpture" (2007)
Pablo Valbuena

Fuente: http://artintelligence.net/review/wp-content/uploads/2008/01/valbuenapablostybd.jpg

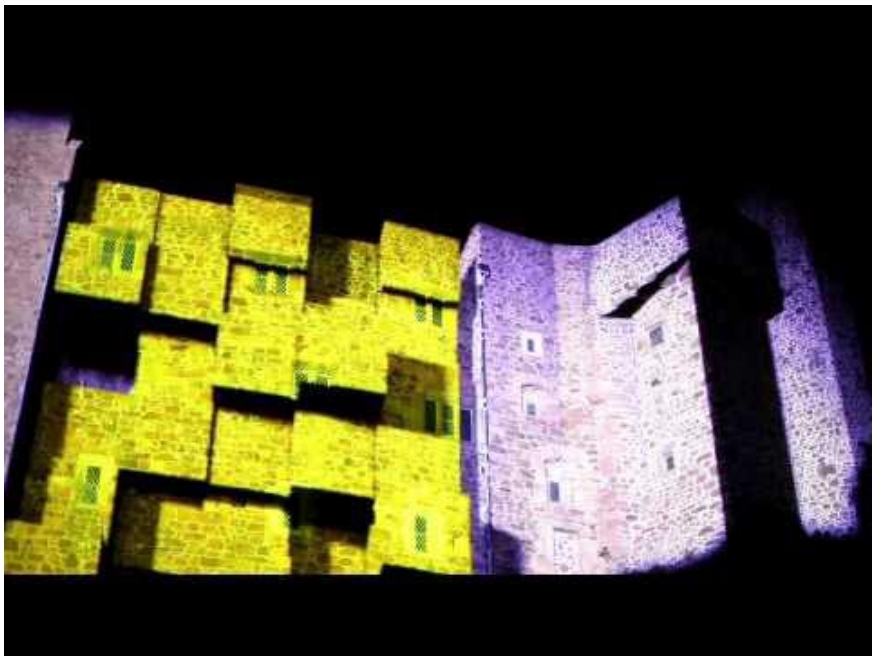


Figura 2

"Battle of Branchage"
(2009)
grupo Seeper

Fuente: https://i1.ytimg.com/vi/zdX4JYsLYtc/maxresdefault.jpg - http://t3.gstatic.com/images?q=tbn:ANd9GcTgnj0PFhqlQfv9U7gCzvbCo_q72pBAQH6kGkkX-fPMg_wMGKM&t=1

Figura 3

"Battle of Branchage"
(2009)
grupo Seeper

Fuente: https://i1.ytimg.com/vi/zdX4JYsLYtc/maxresdefault.jpg - http://t3.gstatic.com/images?q=tbn:ANd9GcTgnj0PFhqlQfv9U7gCzvbCo_q72pBAQH6kGkkX-fPMg_wMGKM&t=1

Figura 4

"Presentación de Peugeot 308"

Radugadesign

(2011)

Fuente: http://www.blogcdn.com/jp.autoblog.com/media/2012/01/audi-video-map.jpg - http://i4.ytimg.com/vi/R1_hLOw1vb4/0.jpg

Figura 5

"Presentación de Peugeot 308"

Radugadesign

(2011)

Fuente: http://www.blogcdn.com/jp.autoblog.com/media/2012/01/audi-video-map.jpg - http://i4.ytimg.com/vi/R1_hLOw1vb4/0.jpg

Aunque mayormente ha sido utilizado en el campo de la expresión artística, el espectáculo y la publicidad, tiene también aplicaciones prácticas como la previsualización de un diseño en producción, o una representación más intuitiva de alguna información.

El problema principal que plantea una representación mapeada es que la imagen virtual, al incidir de manera no perpendicular sobre la superficie a mapear, sufre una deformación a causa de que los haces de luz recorren distancias diferentes desde la fuente hasta su destino.

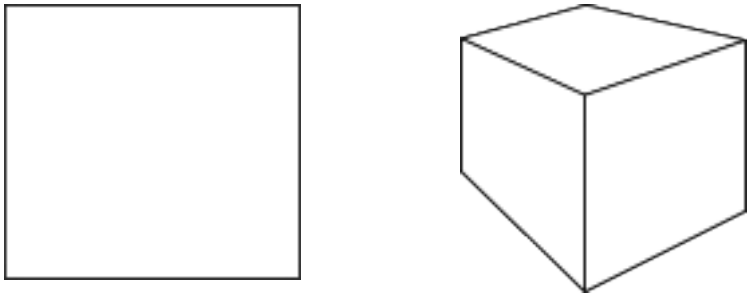


Figura 6 - 7

Izquierda: Plano continuo bidimencional

Derecha: Planos múltiples adyacentes

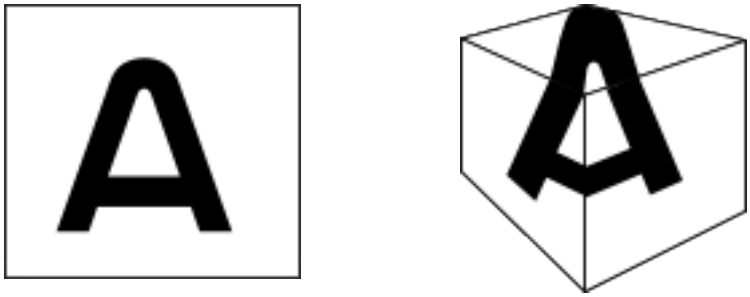


Figura 8 - 9

Izquierda: Proyección de una imagen sobre un plano

Derecha: Proyección de una imagen sobre un cuerpo

La solución es sencillamente contrarrestar esta deformación óptica con una deformación inversamente proporcional sobre la imagen emitida.

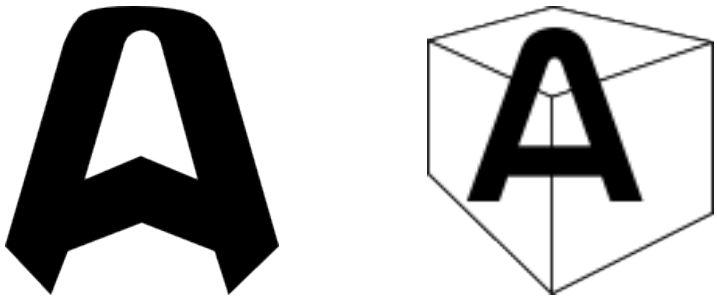


Figura 10 - 11

Izquierda: imagen emitida

Derecha: proyección resultante

Para llevar a cabo esto, hay dos caminos generales. Uno de ellos consiste en la creación de un modelo virtual o físico del objeto que sirva como soporte a la proyección, mientras que el otro consiste en calibrar las deformaciones que afectan la proyección a mano.

Consideraciones generales del contenido

A la hora de planificar un contenido, sin importar que metodologías se utilicen para realizarlo, es deseable tener claro qué nivel de detalle tendrá la deformación compensatoria y que tan estrecha será la relación semántica entre el objeto físico y la imagen virtual.

Siempre será más rico mostrar imágenes que produzcan un diálogo interesante entre los dos aspectos de la representación y siempre producirá una ilusion mas fuerte cuando todos los detalles que tiene la superficie proyectada estén mapeados.

Sin embargo, cada proyecto tiene requerimientos diferentes y es necesario una consideración particular sobre la interacción entre estos elementos.

Otro aspecto que hay que tener en cuenta respecto a la producción de contenido, es definir que elementos del mapeo serán predefinidos y cuales estarán sujetos a variaciones durante la ejecución del mismo. Normalmente cada proyecto tendrá una configuración diferente, dependiendo de varios factores: la escala del mapeo, la posición y forma del objeto a proyectar, la intención de la representación, la disposición de la fuente de proyección, el nivel de interacción posible, etc.

Contenidos predefinidos

Los elementos predefinidos del contenido son aquellos que son creados previamente al mapeo en sí. En la mayoría de los casos donde la imagen y el objeto tienen una fusión fuerte, suele haber una gran cantidad de elementos predefinidos, sobre todo un modelo ya sea virtual o físico del objeto a mapear.

Cuanto más cantidad de elementos estén predefinidos, ciertas características se verán beneficiadas, como el nivel de detalle que puede alcanzar la representación, la ilusión de realidad aumentada, predictibilidad del resultado, los requerimientos técnicos necesarios durante la puesta, etc. A su vez, se reduce la cantidad de interactividad/variabilidad posible en el contenido del proyecto, aumenta la dificultad de solucionar problemas inesperados durante el montaje y ejecución, etc.

Contenido dinámico

Los elementos dinámicos del contenido son aquellos que se generan o modifican durante la ejecución del mapeo. Estos elementos son los que permiten a un proyecto tener mayor o menor variabilidad, abriendo la posibilidad a la interactividad con estímulos producidos por usuarios, bases de datos o cualquier otra entrada de información, en tiempo real. Estos elementos son capaces de dar una gran flexibilidad al proyecto, permitiendo que pueda variar en

vivo tanto la imagen virtual como el objeto físico, y dando mayor posibilidades para resolver distintos problemas que puedan surgir durante una puesta. Sin embargo, cuantos más elementos dinámicos tiene el contenido, el tiempo de montaje y las necesidades de equipamiento técnico suelen ser considerablemente mayores, además de dificultar la cantidad de detalle que puede tener la relación del aspecto virtual con el físico. Esto significa que el contenido dinámico permite la generación o modificación en tiempo real de ciertos aspectos del mapping.

Objetivo

La intención del emmeLab consistía en crear una obra que diera la oportunidad de explorar el uso de video-mapping para poder reconocer y estudiar sus principales problemas y posibilidades. Habiendo observado que la mayoría de los casos de video-mapping trabajaban con videos creados de forma previa al montaje, ejecutados de forma autónoma, se propuso desarrollar un trabajo que incluyera tanto video-mapping, como interactividad e imágenes generadas en tiempo real.

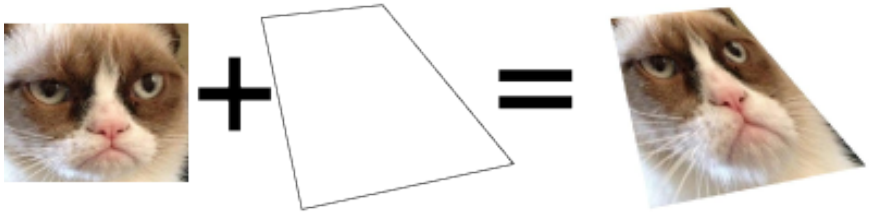
También se decidió usar como objeto de proyección una escultura creada por el emmeLab específicamente para este caso, aprovechando las ventajas que esto conlleva; era más fácil e interesante generar una estética si se tiene el control tanto del aspecto virtual como el fisico, también la escultura estaría en completa disposición para cualquier prueba durante el proceso de producción, así como para cualquier modificación necesaria que surgiera.

Para la creación de las imágenes que caerían sobre la escultura se decidió trabajar con arte generativo pues este permitirá un interesante dinamismo siempre contrastando sobre la firmeza del objeto fisico. Arte generativo se refiere a una pieza de arte que ha sido creada en parte o completamente por un sistema autónomo que desliga al autor de cualquier decisión directa durante su desarrollo.

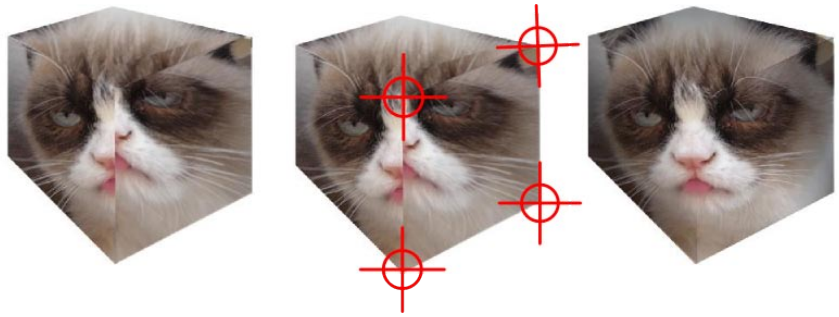
Proceso

Para las pruebas iniciales se usó como soporte de proyección dos cubos de telgopor. Para el software que creará las imágenes se utilizó Processing, un lenguaje de programación abierto y gratuito basado en java ampliamente utilizado en el campo de las artes digitales por ser fácilmente abordable y tener un amplia comunidad activa que colabora en su crecimiento.

Lo primero que se hizo fue texturizar un polígono por cada cara de los cubos. Esto es, tomar una imagen digital y aplicarla sobre un polígono de manera que la imagen digital puede verse estirada, recortada y/o repetida.

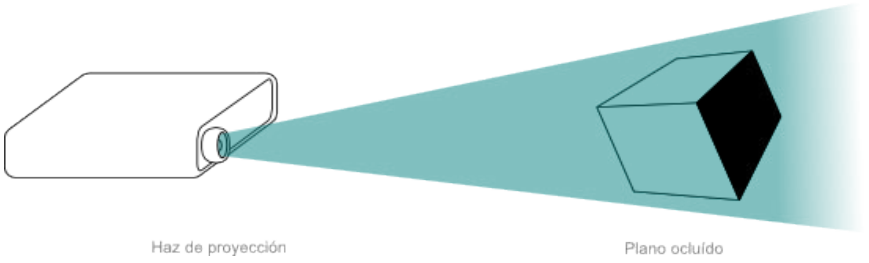


La imagen era generada automáticamente en el código del programa. Luego se proyectaba sobre el cubo los polígonos texturados, y se movían los vértices de las figuras proyectadas hasta hacerlos coincidir con los vértices de las caras físicas del cubo.



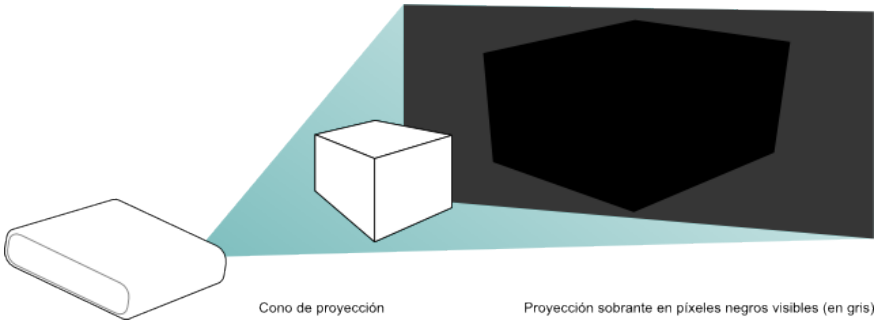
En este punto se observaron varias situaciones.

La luz que emite el proyector tiene una dirección dada, esto significa que hay caras que quedarán ocluídas. Es imposible que se envuelva un objeto enteramente desde un único punto de emisión.

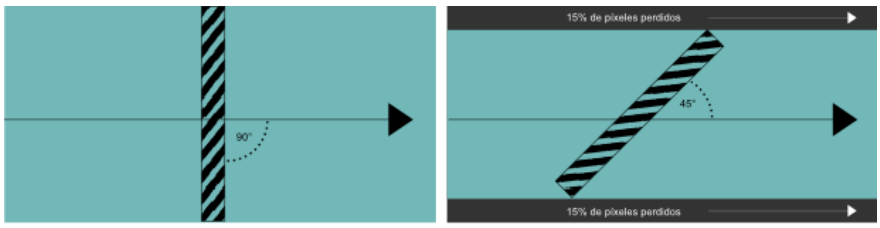


El proyector debe colocarse de manera que el cono de proyección abarque todo el objeto a proyectar, o al menos la zona que se desea mapear, esto causa que gran cantidad de la proyección caiga por fuera del objeto.

Esto significa que gran parte de la proyección es desperdiciada, e incluso en algunos casos puede ser problemático que se vea parte de la luz del proyector (incluso emitiendo píxeles negros se suele observar un rectángulo de luz gris oscuro) más allá de los límites del objeto proyectado.



La dirección de la luz tiene otro efecto sobre la proyección. Un ángulo de incidencia totalmente perpendicular sobre un plano proyectado es óptimo, pero a medida que el ángulo es menor y la el haz se vuelve más rasante, menos pixeles caen sobre una misma cantidad de superficie física, causando que cada cara del cubo tenga diferentes niveles de resolución en su imagen.



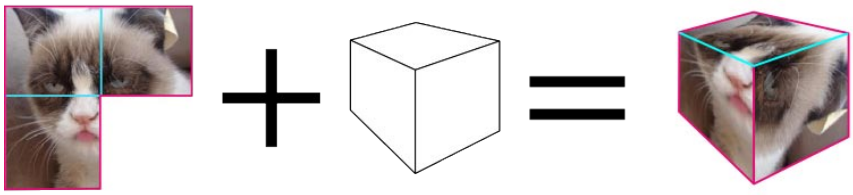
Haz de proyección perpendicular
100% de la cantidad de píxeles aprovechados

Haz de proyección tangencial
Porcentaje de píxeles perdidos (ejemplo a 45°)

Las imágenes utilizadas para cada textura son independientes entre si, esto vuelve imposible generar una continuidad entre las diferentes caras del cubo y atenta contra la ilusión de una textura envolvente propia del objeto.

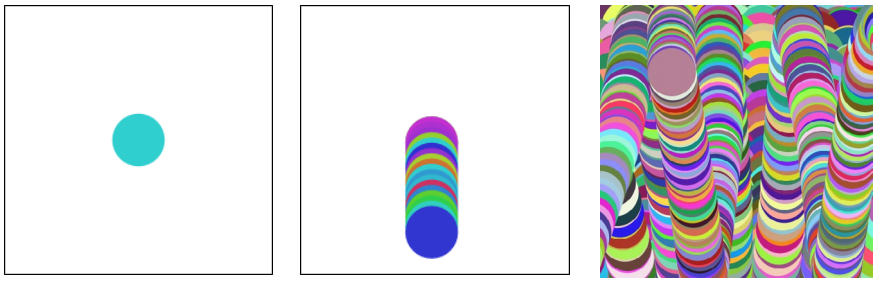
La solución obvia al problema de la oclusión consiste en tener un proyector extra colocado de manera que alcance con su luz las caras que el primero no pudo iluminar. Pero también se encontró que en algunas circunstancias podría generarse un punto de emisión de imagen extra, sin la necesidad de otro proyector: Por un lado los trapezoides creados para realizar el mapeo tienen que estar en coincidencia con las caras del objeto correspondientes, pero por otro lado no necesitan tener una relación relativa entre ellos, es decir que los trapezoides pueden estar en cualquier lugar de la imagen emitida siempre y cuando al caer sobre el objeto coincida con el objeto fisico. Teniendo en cuenta esto, si se coloca un espejo más allá del objeto, redirigiendo aquellos haces que sobran del cono de proyección, se puede redirigir estos pixeles perdidos y hacerlos caer en las caras ocluídas.

La unificación de las texturas era el más complejo e interesante de los anteriores problemas, pues es una fuerte restricción a la hora de generar texturas, y si no se soluciona, no puede esperarse que las imágenes generadas en tiempo real puedan interconectar una cara y otra de forma coherente.

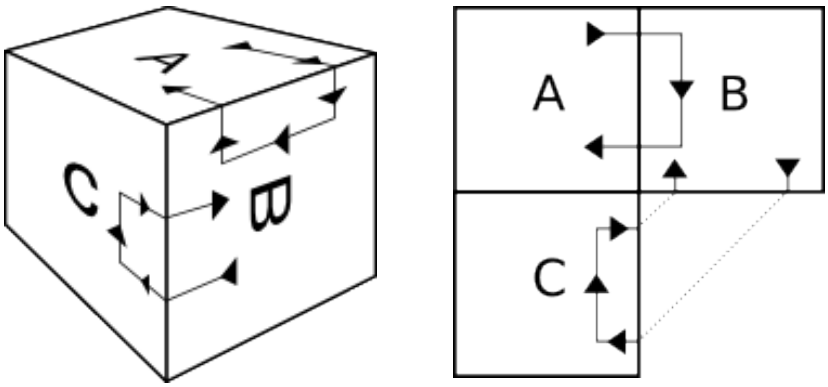


Otros programas que utilizan el mismo modo de polígonos texturizados para compensar las deformaciones no parecen abordar el problema, pues en general solo funcionan para mostrar las imágenes mapeadas, pero no generarlas. La intención de la investigación era crear una plataforma unificada, que permitiera calibrar y producir imágenes, sin necesidad de tener un modelo predeterminado del objeto a proyectar, manteniendo la coherencia de la superficie entera.

Uno de los métodos más comunes para generar arte generativo visual consiste en tener uno o más elementos (llamados agentes) dibujando acumulativamente en un marco dado, dejando que con el tiempo se vaya generando una textura que cambia continuamente hasta que los agentes sean detenidos.

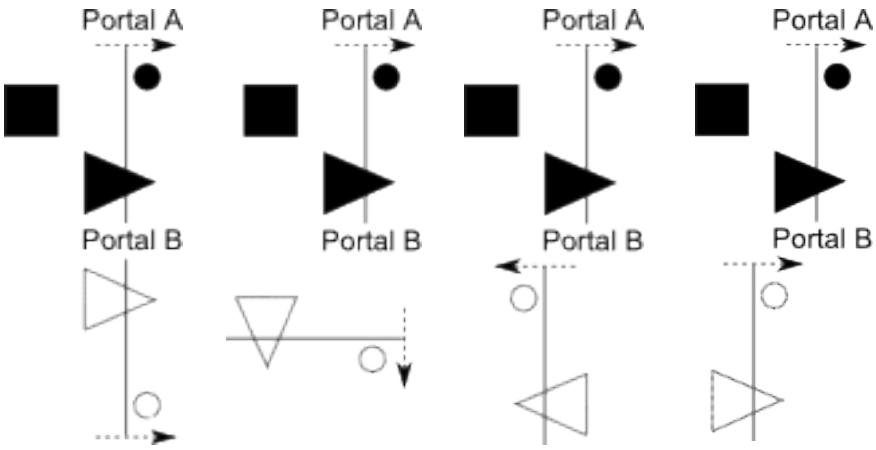


Como era necesario poder generar este tipo de textura para la proyección, hubo que encontrar una forma para que los agentes pudieran recorrer toda la superficie como un solo espacio, debían poder pasar de una cara a cualquier cara adyacente de forma coherente.



Aproximación sobre el software

La primera solución que se planteó fue diseñar una clase (un objeto abstracto de programación) de processing que representaría las aristas del soporte proyectado en forma de segmentos que funcionarían como si fueran pares de portales. Es decir que cualquier elemento gráfico que tuviese un vector de movimiento y cruzara un portal vería modificadas su posición y velocidad de forma relativa a la relacion que habia entre un portal y su par.



Estas aristas virtuales fueron programados con suficientes propiedades para abordar el problema. Poseían dos vértices que definen su posición, rotación y tamaño, uno de estos vértices era marcado como principal y tenía un vector que indicaba su normal. Estos pares de portales poseían una función que permite saber cuando un vector de movimiento en una posición dada lo cruzaría, y era capaz de devolver la posición y vector resultante al "transportarse" a su portal asociado.

Tras varias pruebas se percibió que no solo los portales debían funcionar cuando un elemento los cruzara. Era necesario que cuando un elementos estaba suficientemente cerca como para que parte del mismo lo tocara, había que redibujar (duplicando) el elemento en el otro extremo del portal. Para esto se creo una funcion que además de tomar como argumento la posición, también se indicaba un diámetro, generando en definitiva un circulo abstracto alrededor de cada elemento visual (sin importar la forma de este elemento) que indicaba cuando era necesario registrar.

Aunque el sistema funcionaba, era inadecuado. La configuración era confusa, incluso en un objeto tan sencillo como un cubo, y los cálculos que debía realizar la computadora aumentaban en cantidad demasiado rápido, pues cada elemento debía ser comparado con cada portal existente, entonces no podía esperarse un buen rendimiento y calidad durante una ejecución en tiempo real.

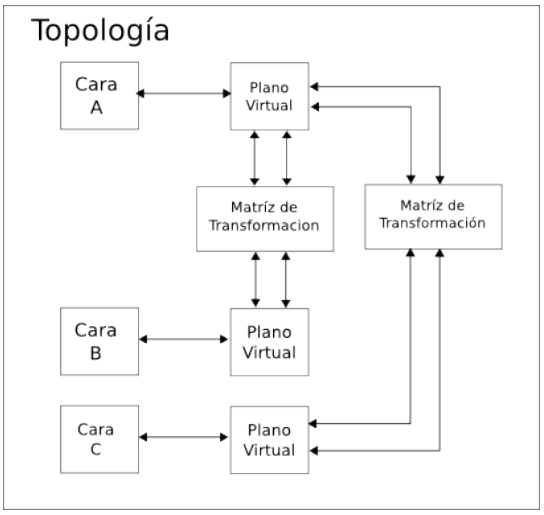
Era necesario plantear una nueva forma de resolver el problema de continuidad.

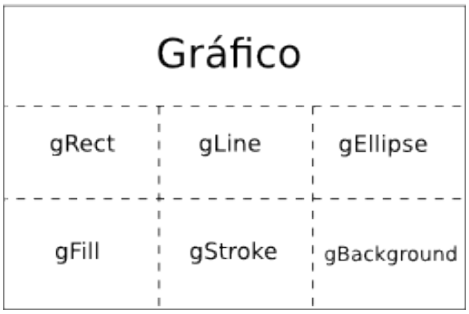
Sistema topológico

Se propuso un nuevo sistema, orientado más hacia cada plano de proyección que hacía las aristas. Cada cara del objeto tendría una representación virtual independiente, pero con información que indicase cuáles eran sus caras adyacentes y en que posición relativa estaban.

De esta manera se podría decir que hay una percepción diferente del resto de las superficies por cada cara. Así, sin importar la cara en la que se use de referencia, se puede tener una representación adecuada de la topología de los planos de proyección.

Para llevar a cabo esta solución se creó un sistema con diferentes clases interrelacionadas.





la “Topología” no es mucho más que un administrador para las diferentes instancias de “Cara” y “Plano Virtual”. Sus funciones permiten recorrer fácilmente todos los elementos que contiene para ejecutar las actualizaciones o realizar alguna modificación a todos ellos.

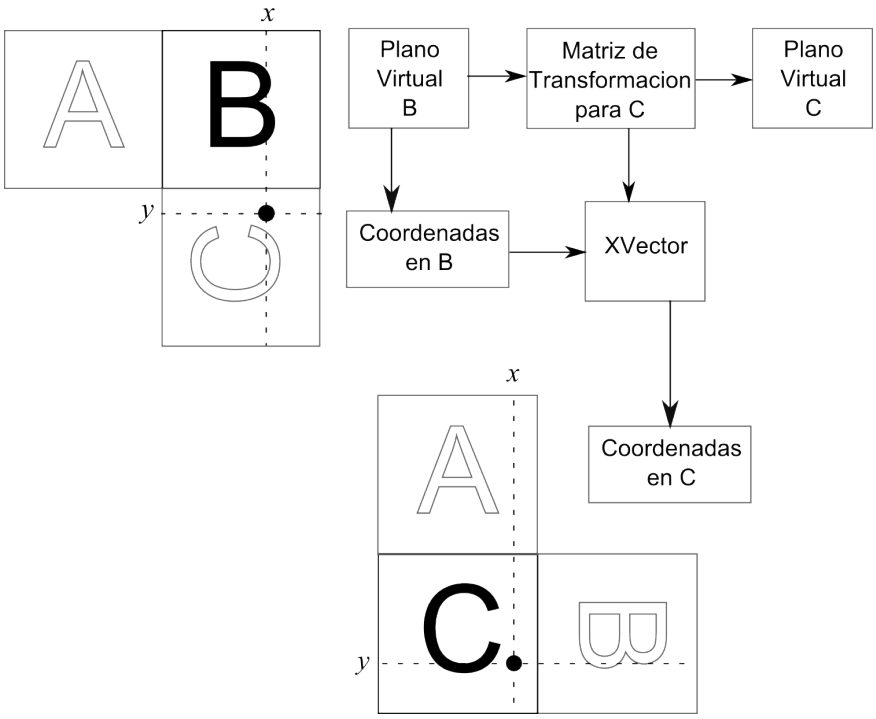
Cada cara proyectada del objeto físico tiene una representación en la forma de un objeto “Cara” y un “Plano Virtual” que están interconectados entre sí. La “Cara” virtual tiene la información de forma para dibujar el trapezio que se proyecta, mientras que el “Plano Virtual” contiene la imagen que se utiliza como textura para este trapezio. El “Plano Virtual” tiene también indexados todos los Planos Virtuales que sean adyacentes junto a las Matrices de Transformación que describen su posición relativa.

La “Matriz de transformación” contiene información de rotación y posición que indican cómo se relacionan las caras adyacentes entre sí.

Estas clases se combinan con las clases “XVector”, “Gráfico” y sus extensiones, para resolver las diferentes necesidades que surgen durante la generación de las imágenes de forma accesible.

La clase “XVector” tiene la función de permitir trabajar con posiciones y ángulos, sin que se pierda la coherencia de la topología entera. Ya sea para mover un punto usando un vector de velocidad, o para calcular la distancia entre dos puntos que estén en diferentes caras, este objeto tiene la función de darnos un resultado que respete la configuración que posee la superficie de proyección. Para utilizar las funciones de “XVector” se crea inicialmente una instancia temporal que usa como datos iniciales una coordenada cartesiana y una “Plano Virtual” como referencia de origen. Luego existen una serie de funciones que devuelven el mismo XVector con la información resultante de la operación realizada:

- movPolar : Como argumentos recibe una distancia y un ángulo. Toma la coordenada inicial y se la desplaza utilizando el vector polar recibido. Luego calcula la nueva coordenada cartesiana, y en caso de haber pasado a una nuevo plano, transforma la coordenada para ese nuevo plano indicando además cuál es.
- puntoDeMiACara : Toma una coordenada cartesiana desde el plano de origen, y un plano de destino como argumentos, entonces devuelve las coordenadas cartesianas de ese mismo punto, pero desde la perspectiva del plano de destino.
- puntoDeCaraAMi : La inversa de “puntoDeMiACara”, el punto calculado usa como referencia el plano pasado como argumento y lo traspasa al plano de origen del XVector.
- caraDePunto : toma una coordenada como argumento y usando como referencia el plano de origen del XVector, devuelve el Plano Virtual que contendrìa ese punto.

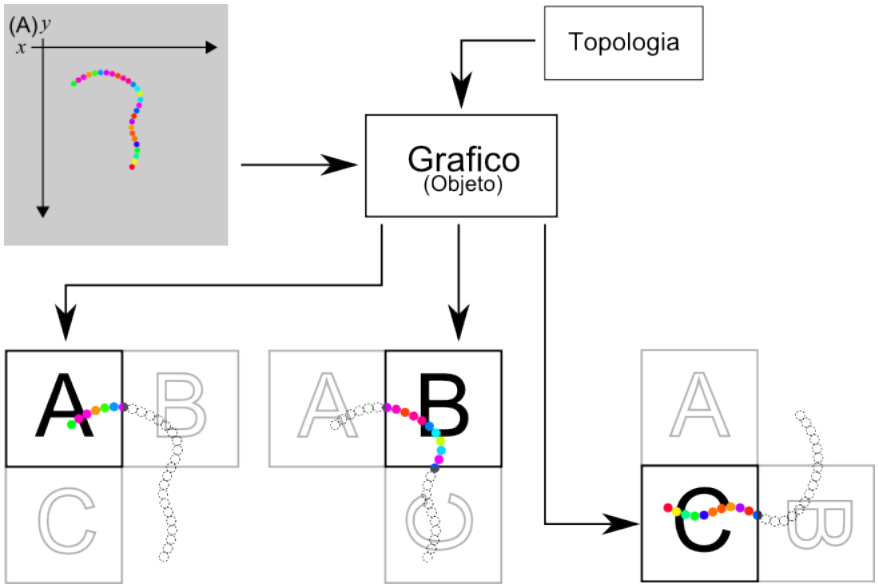


Este esquema representa el uso de “puntoDeMiACara” donde el punto de origen es una coordenada en el plano B, y se usa al XVector para que calcule ese mismo punto pero desde la perspectiva del plano C

La clase “Grafico” funciona como una interfaz entre la Topología y las funciones de dibujo de Processing. Cada una de estas funciones tiene un objeto que es una extensión de “Grafico” como por ejemplo “gLine” o “gRect”. Estos objetos permiten que el usuario pueda dibujar sobre la topología de la misma forma en que dibujaría sobre un applet de Processing normal con la única excepción que debe indicar la cara que se esta utilizando como referencia.

Básicamente cada una de las extensiones de la clase “Grafico” tienen dos funciones: una es “caraTocada” utilizada por el programa para calcular que texturas serían afectadas por la función de dibujo, y la otra es “forma” que posee las instrucciones que reproducen el gráfico en si.

El proceso que se lleva a cabo cada vez que se utiliza uno de estos objetos al dibujar, consiste básicamente en un recorrido por el Plano Virtual indicando como referencia y por todos los Planos Virtuales que sean adyacentes. Por cada uno de ellos que sea considerado “tocado” por el gráfico, se ejecuta la “forma” realizando antes las transformaciones necesarias de los valores geométricos que definen el dibujo (como los vértices en un triángulo, o el centro en una elipse).



Sin embargo, aunque este proceso parece complicado, el usuario del software no necesita más que llamar tan solo una de las funciones de dibujado basadas en las propias de Processing (por ejemplo “gEllipse”) indicando antes cual es la cara que debe usarse como referencia, del mismo modo que se indica el color de relleno antes de dibujarse un polígono.

Producción de la pieza

Cuando el software estuvo terminado, se pudieron tener en cuenta las ventajas y limitaciones que proporcionaba y se comenzó el desarrollo de la pieza específica con la que se implementaría el sistema.

Los requerimientos necesarios para responder a la intención inicial estaban resueltos; el sistema soportaba la capacidad de mapear una escultura y de generar una textura en tiempo real que la cubriera, pero el software presenta una limitación: las texturas generadas tendrán un marco rectangular necesariamente, es decir que el objeto debía tener caras rectangulares para funcionar de forma óptima con el programa.

Esto nos llevó a una escultura con un estilo semejante al del movimiento arquitectónico metabolista, con características estructurales que sugerían rigidez, ya que en este caso el componente orgánico propio del movimiento arquitectónico lo aportaría el comportamiento de la imagen proyectada.

Sin embargo, a pesar de surgir a causa de una limitación del sistema, las cualidades de la escultura eran un buen refuerzo semántico al concepto de la escultura como el aspecto físico, concreto y definido de la obra, mientras generaba un fuerte contraste con su contrapartida virtual, intangible y potencial.

Para definir la interacción en la obra se propuso una operación de usuario que continuó por esta línea. Se dio al usuario el rol de

“animar” la escultura, usando del soplido o la voz como forma de ingresar estímulos al sistema. La idea era que el sonido o el viento, a pesar de pertenecer al reino físico, puede asociarse con ciertas cualidades del reino virtual, como la impermanencia, la intangibilidad, funcionando de alguna manera como puente entre el aspecto virtual y el aspecto físico de la obra.

La escultura presenta algunas zonas sensibles al usuario, desde donde una textura se propagaron al resto de la escultura al detectar cualquier soplido o sonido que generase el usuario cubriendo zonas inactivas o chocando las propagaciones generadas por otros usuarios. Desde el momento que el usuario deja de estimular al sistema, las texturas comenzaron a perder impulso, regresando poco a poco al estado de pasividad inicial, aunque dejando rastros de su paso sobre la escultura que solo se verán modificadas al ser cubiertas por una nueva textura.

Fue necesario resolver una serie de problemas técnicos, más allá del video-mapping, para llevar a cabo la propuesta, principalmente la forma en que sería sensada la voz y el soplido de los usuarios.

Sensado de sonido

El sensado de sonido para Mopp fue resuelto mediante cuatro micrófonos electret, un circuito electrónico de amplificación desarrollado específicamente para este proyecto, una placa de microcontrolador Arduino, un puerto de comunicación inalámbrica Xbee y software escrito en Processing.

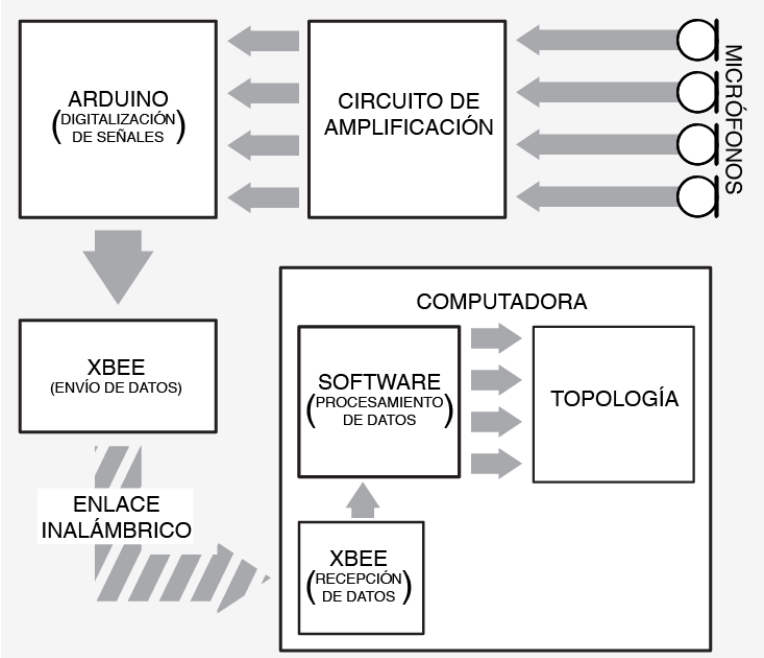
Los micrófonos están posicionados en las cuatro caras de la escultura, ocultos bajo su superficie.

El circuito amplificador alimenta y amplifica los micrófonos. Pasada la etapa de amplificación, la señal resultante de cada micrófono es transmitida de manera independiente a cuatro conversores analógico/digital de la placa Arduino.

La plataforma Arduino muestrea el valor de amplitud de cada una de las señales y las envía mediante comunicación serial (a través del puerto Xbee) a la computadora.

El software que recibe los datos utiliza un sistema de rampa para filtrar ruido en la señal y un parámetro de umbral para determinar si alguno de los micrófonos debería considerarse “activado”. La condición de activo/inactivo de cada micrófono es utilizada para disparar las animaciones que cubren la escultura, partiendo desde cada uno de los puntos bajo los cuales están ubicados los micrófonos.

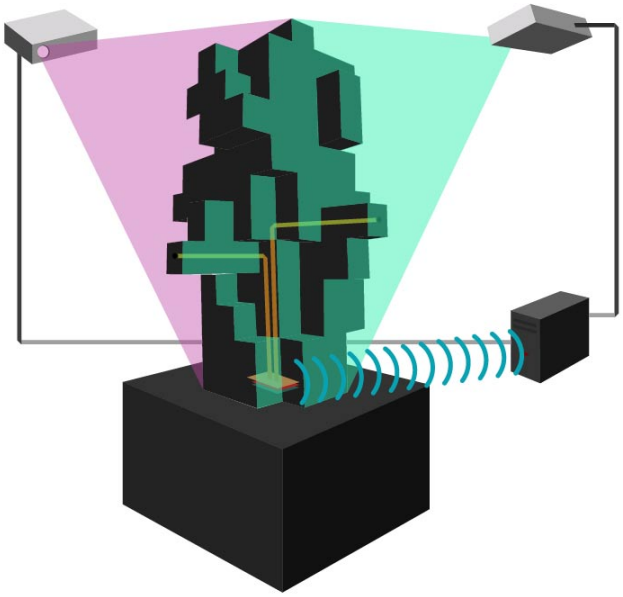
Este esquema representa la totalidad del sistema de sensado de sonido, y su vinculación con el software de mOpP.



Montaje de la obra:

Dadas las dimensiones de la pieza escultórica y la necesidad de que la proyección fuese capaz de envolverla casi por completo, fue necesario el uso de dos proyectores que atacaran el volumen desde ángulos opuestos. Estos se encontraban conectados a una computadora que generaba una imagen de 2048 x 768 pixels dividida entre los dos proyectores. El sistema de sensado de sonido fue oculto dentro del volumen a proyectar, y esto junto con el enlace inalámbrico permitieron sellar la escultura de manera que el equipamiento electrónico no influyera en el aspecto visual de la misma.

Esta ilustración muestra un esquema del montaje completo de la obra, con todos los elementos que intervienen en su funcionamiento.



Calibración del video-mapping

Durante las etapas iniciales del desarrollo del software, la calibración de los polígonos compensatorios se hacía entrando en un modo especial de configuración en el propio programa. Como esta modalidad era extremadamente tediosa e ineficiente, y la intención de la investigación era generar una herramienta al alcance de todos, se comenzó a desarrollar un sistema aparte para la calibración.

En una primera instancia se reemplazó el modo especial de configuración por un simple sistema de carga e interpretación de archivos XML, de manera que cualquiera pudiese desarrollar un programa capaz de generar la calibración que utilizaba la Topología. Luego se empezó a generar un programa a parte, que evolucionaba constantemente en cada prueba de proyección que se hacía sobre la escultura física.

De esta manera el programa de configuración acabó siendo muy versátil, con suficientes funcionalidades para resolver las problemáticas que se encontraban durante la etapa de calibración del video-mapping.

Conclusión

Con mOpP se intentó abordar el problema del video-mapping desde una perspectiva relativamente inexplorada, diseñando una plataforma de desarrollo que abre la posibilidad de generar contenidos independientes del modelo físico proyectado.

Este proyecto demuestra que aún queda mucho terreno por explorar en el ámbito del video-mapping, y que es necesario continuar el desarrollo de herramientas que faciliten el uso de este recurso.

El software de mOpP tiene el potencial, dada sus características, de ser expandidos y de asociarse con diferentes proyectos dedicados a la producción audiovisual interactiva. Es importante valorar la postura de código abierto que posee el proyecto, pues esto es lo que permite crecer a esta clase de herramientas, más allá de sus alcances iniciales.

Fuentes de internet

Processing:

Disponible en <http://www.processing.org>

Projection mapping:

Disponible en <http://www.projection-mapping.org/>

Pure data:

Disponible en <http://www.puredata.info/docs>

vvvv:

Disponible en <http://www.vvvv.org/>

Max-MSP:

Disponible en <http://cycling74.com/>

Paintingwithlight:

Disponible en <http://paintingwithlight.bigfug.com/>

// Notas

1. El Emmelab es un Laboratorio de investigación y experimentación en nuevas interfaces para el arte, dependiente del Departamento de Multimedia de La Facultad de Bellas Artes, UNLP.