

Proyecto mOpP

UNA HERRAMIENTA PARA PROYECCIONES DE VIDEO SOBRE UN VOLUMEN¹

Francisco Álvarez Lojo
Ezequiel Rivero
Ariel Uzal

// Laboratorio Emmelab², Departamento de Multimedia, Facultad de Bellas Artes,
Universidad Nacional de La Plata.

Resumen

Gracias a la disponibilidad de equipos tecnológicos avanzados y a la proliferación de comunidades y de foros en internet, el *video-mapping* se utiliza cada vez con mayor frecuencia. Sin embargo, existen pocas herramientas que tengan en cuenta los problemas que suelen presentarse durante el desarrollo de una pieza de este tipo.

Frente a esto, el grupo Emmelab comenzó a realizar un trabajo que utiliza *mapping* y un *software* capaz de abordar sus principales problemas. El resultado del proyecto, que se describe en este artículo, se denominó mOpP, y consta de una escultura envuelta por una proyección de imágenes interactivas generadas en tiempo real.

Palabras clave

Mapping - Proyecciones volumétricas - Realidad aumentada - *Processing*

En estos años hemos visto utilizar, cada vez con más frecuencia, el recurso visual denominado *video-mapping*. Esta técnica es una manifestación de realidad aumentada, cuya principal intención 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.

El problema que plantea una representación mapeada es que la imagen virtual, al incidir de manera no perpendicular sobre la superficie que debe mapearse, sufre una deformación de naturaleza óptica [Figura 1].

La solución es contrarrestar la deformación óptica con una deformación inversamente proporcional sobre la imagen emitida [Figura 2].

Para llevar a cabo este proceso hay dos caminos: uno consiste en crear un modelo virtual o físico del objeto, que sirva como soporte para la proyección; el otro, que fue el seguido en la investigación, consiste en calibrar manualmente las deformaciones que afectan a la proyección.

Otro de los problemas que presenta el *video-mapping* es que al producir los contenidos que serán mapeados se debe atender una serie de consideraciones que



varían según las características de cada proyecto. A la hora de planificar un contenido, sin importar las metodologías que se utilicen para realizarlo, es deseable saber qué nivel de detalle tendrá la deformación compensatoria y cuán 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 la ilusión será más fuerte si los detalles que tiene la superficie proyectada están mapeados.

Los elementos predefinidos del contenido son los que se crean antes de hacer el mapeo. En la mayoría de los casos, en los que la imagen y el objeto tienen una fusión fuerte, suele haber una gran cantidad de elementos predefinidos, sobre todo un modelo virtual o físico del objeto que se quiere mapear.

Los elementos dinámicos del contenido son aquellos que se generan o que se modifican durante la ejecución del mapeo; son los que permiten que un proyecto tenga mayor o menor variabilidad, y los que posibilitan la interactividad con estímulos producidos por usuarios, las bases de datos o por cualquier otra entrada de información en tiempo real. La flexibilidad que aportan estos elementos radica en que permiten, en vivo, variar tanto la imagen virtual como el objeto físico, y en que brindan más posibilidades para resolver los problemas que puedan surgir durante una puesta.

Desarrollo del Proyecto mOpP

La intención del Emmelab fue crear una obra que permitiera explorar el uso del *video-mapping* para reconocer y para estudiar sus principales problemas y sus posibilidades expresivas. El resultado del Proyecto de Investigación, que se denominó mOpP, consta de una escultura completamente envuelta por una proyección de imágenes interactivas generadas en tiempo real.

Al observar que en la mayoría de los casos de *video-mapping* se trabajaba con videos creados de manera previa al



Figura 1. Proyección de una imagen sobre un plano y sobre un cuerpo



Figura 2. Imagen emitida y proyección resultante

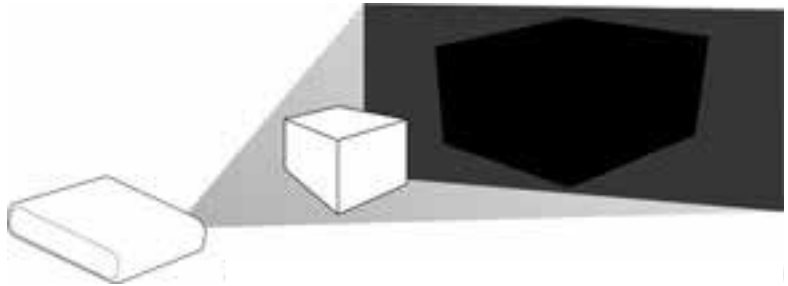


Figura 3. Volumen abarcado completamente por la proyección



Figura 4. Volumen con una sola fuente de proyección

montaje y ejecutados en forma autónoma, se propuso desarrollar un trabajo que incluyera tanto el *video-mapping* como la interactividad y las imágenes generadas en tiempo real. También se decidió usar como objeto de proyección una escultura a la que se la llamó mOpP, creada por el EmmeLab especialmente para este caso, para aprovechar las ventajas que esto conlleva: es más fácil e interesante generar una estética si se tiene el control del aspecto virtual y del aspecto físico. Esta escultura, además, podía estar disponible para cualquier prueba durante el proceso de producción y para cualquier modificación que fuera necesaria.

Para construir las imágenes que se proyectaron sobre la escultura se decidió trabajar con arte generativo⁴, puesto que permite un interesante dinamismo y contrasta sobre la firmeza del objeto físico. El software con el que se crearon las imágenes fue *Processing*⁵.

En las pruebas iniciales, se usaron como soportes para la proyección dos cubos blancos de 20 cm de lado. En primer lugar, se texturizó un polígono por cada cara de los cubos; es decir, se tomó una imagen digital y se la aplicó sobre un polígono, de manera que pudiera verse estirada, recortada y/o repetida. Como resultado, la imagen fue generada automáticamente en el código del sistema mOpP. Luego, se proyectaron sobre el cubo los polígonos texturados y se movieron los vértices de las figuras proyectadas hasta hacerlos coincidir con los vértices de las caras físicas del cubo.

Como la intención era cubrir el objeto por completo, comprobamos que el proyector debía colocarse de manera que el cono de proyección abarcara todo el objeto que se quería proyectar o, al menos, la zona que se deseaba mapear [Figura 3]. Esto causó que buena parte de la proyección quedara por fuera del objeto, y por



Figura 5. Dos superficies del mismo tamaño ocupan menos píxeles a causa del ángulo de incidencia

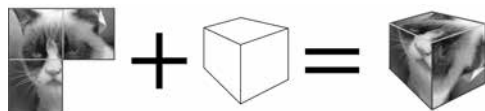


Figura 6. La misma imagen es continua, desenvuelta en un plano, pero no en un volumen

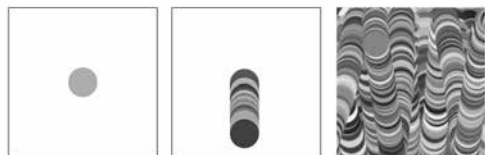


Figura 7. Un solo elemento genera una textura a partir del movimiento y de la repetición

lo tanto, que una parte importante sea desperdiciada. En algunos casos, incluso, puede ser problemático que se vea parte de la luz del proyector –si se emiten píxeles negros también se suele observar un rectángulo de luz gris oscuro– más allá de los límites del objeto proyectado.

Otra consecuencia fue que la luz que emitía el proyector tenía una dirección dada, lo que quiere decir que había caras del cubo que quedaban ocluidas [Figura 4]. Esto sucedió porque es imposible que un objeto se envuelva, por completo, desde un único punto de emisión. La solución obvia al problema de la oclusión consistía en tener un proyector extra colocado de manera que alcanzara con su luz las caras que el primero no podía iluminar. Entonces, los trapezoides podrían estar en cualquier lugar de la imagen emitida siempre y cuando, al caer sobre el objeto, coincidieran con el objeto físico. De este modo, si se colocara un espejo más allá del objeto, que redirigiera aquellos haces que sobran del cono de proyección, se podrían aprovechar estos píxeles perdidos y caerían en las caras ocluidas.

Además, se observó que la dirección de la luz tenía otro efecto sobre la proyección [Figura 5]. Un ángulo de incidencia per-

pendicular sobre un plano proyectado es óptimo para una representación fiel, pero a medida que el haz se vuelve más rasante menos píxeles caen sobre una misma cantidad de superficie física. Esto hace que cada cara del cubo tenga diferentes niveles de resolución en su imagen.

Las imágenes utilizadas para cada textura⁶ eran independientes entre sí, lo que no permitía generar una continuidad entre las diferentes caras del cubo y atentaba contra la ilusión de una textura envolvente, propia del objeto. La unificación de las texturas fue el más complejo e interesante de los problemas, pues es una fuerte restricción a la hora de generar texturas. Si no se soluciona, no puede esperarse que las imágenes generadas en tiempo real puedan conectar una cara con otra de forma coherente [Figura 6].

Otros programas que utilizan el mismo modo de polígonos texturizados para compensar las deformaciones no parecen abordar este problema; en general, sólo funcionan para mostrar las imágenes mapeadas, pero no las generan. Por este motivo, el objetivo del Proyecto fue 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 producir *arte generativo visual* consiste en tener uno o más elementos –llamados “agentes”– que dibujen, acumulativamente, en un marco dado, y dejar que, con el tiempo, se genere una textura que cambie continuamente hasta que los agentes se detengan [Figura 7].

Aproximación al software

Para generar una textura que cambiara continuamente, se buscó que los agentes pudieran recorrer toda la superficie como si fuera un solo espacio, pasando de una cara a cualquier otra cara adyacente de manera coherente [Figura 8].

La primera solución que se planteó fue diseñar un objeto de *Processing* que representara, en forma de segmentos que funcionaran como si fueran pares de portales⁷, las aristas del soporte proyectado. Pero aunque funcionaba, el sistema era inadecuado: la configuración era confusa, incluso para un objeto tan sencillo como un cubo, y los cálculos que debía realizar la computadora aumentaban demasiado rápido, pues cada elemento debía ser comparado con cada portal existente. Por este motivo, no podía esperarse un buen rendimiento y una buena calidad durante una ejecución en tiempo real.

Para resolver el problema de la continuidad, se propuso un nuevo sistema, orientado más hacia cada plano de proyección que hacia las aristas [Figura 9]. Cada cara del objeto tendría una representación virtual independiente, pero con información que indicase cuáles eran sus caras adyacentes y en qué posición relativa estaban.

De esta manera, se podría decir que había una percepción diferente del resto de las superficies de cada cara. Por ello, sin importar la cara que se usara como referencia, se podía tener una representación adecuada de la topología de los planos de proyección.

Con diferentes objetos interrelacionados, se creó un sistema que consiste en varios objetos con la información necesaria para describir la superficie del volumen

a la computadora, y dos objetos especiales que tienen las funciones que hacen uso de dicha información [Figura 10].

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 para realizar alguna modificación.

Cada cara proyectada del volumen físico tiene una representación virtual en la forma de un objeto de programación cara y otro plano virtual. La cara virtual posee la información de forma para dibujar el trapecio que se proyecta; mientras que el plano virtual contiene la imagen que se utiliza como textura para este trapecio. El plano virtual tiene indexados, también, todos los planos virtuales que sean adyacentes y las matrices de transformación⁸ que describen su posición relativa.

Se crearon dos objetos, XVector y Gráfico –y las extensiones de éste último: gRect, gLine, gEllipse, gFill, gStroke y gBackground–, que al combinarse con los planos virtuales permiten resolver fácilmente las necesidades que surgen durante la generación de imágenes.

El objeto XVector permite trabajar con posiciones y con 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 dar un resultado que respete la configuración que posee la superficie de proyección. Para utilizar las funciones de XVector se creó, inicialmente, una instancia temporal que usó como datos iniciales una coordenada cartesiana y un plano virtual como referencia de origen. Esta instancia temporal posee una serie de funciones que devuelven el mismo XVector con la información resultante de la operación realizada, éstas son: movPolar, PuntoDeMiCara, puntoDeCaraAMi, caraDePunto.

El siguiente esquema representa el uso de puntoDeMiCara, en el que el punto de origen es una coordenada en el Plano B, y se usa el XVector para que calcule ese

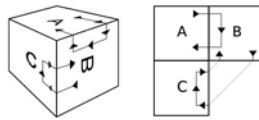


Figura 8. Representación de la continuidad en el volumen pasada a un plano bidimensional



Figura 9. Cada cara del cubo tiene su propia relación espacial con las demás

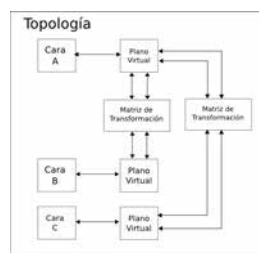


Figura 10. Esquema simplificado de los objetos y sus relaciones en el software creado

mismo punto pero desde la perspectiva del Plano C [Figura 11].

El objeto Gráfico 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 Gráfico, como por ejemplo gLine o gRect. Estos objetos permiten que el usuario pueda dibujar sobre la topología de la misma forma en la que dibujaría sobre un *applet* de *Processing* normal, con la única excepción que debe indicar la cara que está utilizando como referencia.

Cada una de las extensiones de la clase Gráfico tiene dos funciones: una es *cara-Tocada*, utilizada por el Programa para calcular qué texturas serían afectadas por la función de dibujo; la otra es *forma*, que posee las instrucciones que reproducen el gráfico.

El usuario del software sólo necesita llamar una de las funciones de dibujado basadas en las propias de *Processing* –por ejemplo, gEllipse– e indicar, con anterioridad, cuál 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 analizar las ventajas y las limitaciones que proporcionaba y se comenzó a desarrollar 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, en tiempo real, una textura que la cubriera. Sin embargo, el software presentaba una limitación: las texturas generadas tenían un marco rectangular, es decir, el objeto debía tener caras rectangulares para funcionar de forma óptima con el Programa. Esto condujo hacia 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.

A pesar de surgir a causa de una limitación del Sistema, las cualidades de la escultura, como el aspecto físico, concreto y definido de la obra, eran un buen

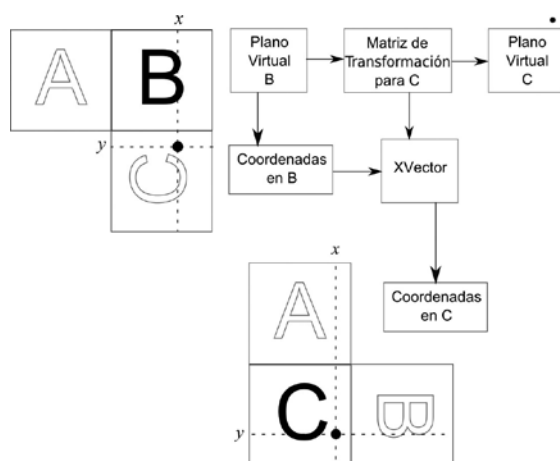


Figura 11. Transformación de las coordenadas de un mismo punto según el plano usado como origen

refuerzo semántico para el concepto de la escultura y, a su vez, generaban un fuerte contraste con su contrapartida virtual, intangible y potencial.

Para definir la interacción en la obra se propuso una *operación de usuario*; es decir, se le dio al usuario el rol de animar la escultura, usando el soplido o la voz como forma de ingresar estímulos al Sistema. El objetivo era que el sonido o el viento, a pesar de pertenecer al reino físico, pudieran asociarse con ciertas cualidades del reino virtual, como la falta permanencia y la intangibilidad, y funcionar como puente entre el aspecto virtual y el aspecto físico de la obra.

La escultura presentaba algunas zonas sensibles para el usuario, desde donde una textura se propagaba al resto de la escultura al detectar cualquier soplido o sonido que generase el usuario y cubría zonas inactivas o chocaba las propagaciones generadas por otros usuarios. Desde el momento en el que el usuario dejaba de estimular al Sistema, las texturas comenzaban a perder impulso y regresaban, poco a poco, al estado de pasividad inicial. No obstante, dejaban rastros de su paso sobre la escultura que solo se verán modificadas al ser cubiertas por una nueva textura.

Para llevar a cabo la propuesta, fue necesario resolver una serie de problemas técnicos, además de los relacionados con

el *video-mapping*. Los principales inconvenientes estuvieron vinculados con la forma en que sería capturado el sonido producido por los usuarios. Para mOpP, la captura de sonido se resolvió 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 el software escrito en *Processing*.

Los micrófonos estaban ubicados en las cuatro caras de la escultura, ocultos bajo su superficie. El circuito amplificador alimentaba y amplificaba los micrófonos. Pasada la etapa de amplificación, la señal resultante de cada micrófono fue transmitida de manera independiente a cuatro conversores analógicos/ digitales de la placa Arduino. Esta plataforma 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.

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 preciso usar dos proyectores que atacaran el volumen desde ángulos opuestos. Éstos se encontraban conectados a una computadora que generaba una imagen de 2048 x 768 píxeles, dividida entre los dos proyectores. El sistema de captura de sonido se ocultó dentro del volumen a proyectar y, junto con el enlace inalámbrico, sellaron la escultura de manera que el equipamiento electrónico no influyera en su aspecto visual.

Durante las etapas iniciales del desarrollo del software, para calibrar los polígonos compensatorios había que utilizar 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 generó un programa separado 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*.

Con mOpP se abordó el *video-mapping* desde una perspectiva relativamente inexplorada, diseñando una plataforma de desarrollo que permite generar contenidos independientes del modelo físico proyectado. El 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 expandido 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 le permite, a esta clase de herramientas, crecer más allá de sus alcances iniciales.

Notas

- 1 El presente artículo se desprende del Proyecto de Investigación "Nuevos territorios de la Generatividad en las Artes Electrónicas: su convergencia con la Robótica, la Realidad Aumentada y el Net.Art", dirigido por el Ing. Emiliano Causa, en el marco del Programa de Incentivos del Ministerio de Educación y Cultura de la Nación (2011-2012).
- 2 El Emmelab es un laboratorio de investigación, análisis y experimentación en nuevas interfaces para el arte formado por docentes y por alumnos de la carrera Diseño Multimedial de la Facultad de Bellas Artes.
- 3 El arte generativo es 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.

- 4 *Processing* es un lenguaje de programación abierto y gratuito muy utilizado actualmente en el campo de las artes digitales.
- 5 Se llamará textura a la imagen digital aplicada al polígono.
- 6 Los Portales, en este caso, son dos segmentos en un espacio bidimensional que generan entre ellos una continuidad espacial.
- 7 La matriz de transformación contiene información sobre rotación y posición que indica cómo se relacionan las caras adyacentes entre sí.

Páginas web consultadas

- <http://gsvideo.sourceforge.net/reference/index.html>
<http://ubaa.net/shared/processing/opencv/opencv.html>
<http://www.generative-gestaltung.de/codes/generativedesing/index.html>
<http://arduino.cc/>
<http://www.sparkfun.com/datasheets/BreakoutBoards/Amplified-Mic-Electret-v14.pdf>