

---

# Recreación de estructuras arquitectónicas mediante modelaje procedural basado en volúmenes

## Recreation of architectural structures using procedural modeling based on volumes

Santiago Barroso Juan y Gustavo Patow

Grup de Geometria i Gràfics, Universitat de Girona, España.

---

### **Resumen**

*Si bien el modelado procedural de edificios y otras estructuras arquitectónicas ha evolucionado de manera muy significativa en los últimos años, existe una notable ausencia de herramientas de alto nivel que permitan a un diseñador, un artista o un historiador, crear edificios o estructuras arquitectónicas importantes de alguna ciudad en particular. En este trabajo presentaremos una herramienta para la creación de este tipo de edificios de manera simple y clara, siguiendo reglas que utilicen el lenguaje y metodología de creación propios de los edificios, y ocultando al usuario los detalles algorítmicos de la creación del modelo.*

**Palabras Clave:** MODELADO PROCEDURAL, ARQUITECTURA, MAMPOSTERÍA, VOLÚMENES, SKYLINE ENGINE.

---

### **Abstract**

*While the procedural modeling of buildings and other architectural structures has evolved very significantly in recent years, there is noticeable absence of high-level tools that allow a designer, an artist or an historian, creating important buildings or architectonic structures in a particular city. In this paper we present a tool for creating buildings in a simple and clear, following rules that use the language and methodology of creating their own buildings, and hiding the user the algorithmic details of the creation of the model.*

**Key words:** PROCEDURAL MODELING, ARCHITECTURE, MASONRY BUILDING, VOLUMES, SKYLINE ENGINE.

---

## 1 INTRODUCCIÓN

La tendencia actual en el modelado procedural de edificios consiste en utilizar técnicas basadas en reglas gramaticales que operan sobre geometría, como se demuestra en los trabajos de Wonka et al. [1], luego mejorado por Müller et al. [2]. Sin embargo, estos métodos sólo permiten un control indirecto sobre el modelo final, principalmente debido a que las reglas que

lo definen se encuentran almacenadas en un archivo de texto que se procesa para obtener el resultado final.

Asimismo, el paradigma actual para el modelado procedural de edificios está enfocado solamente en la fachada de éstos, mientras que usualmente se ignora por completo modelar el interior de los mismos [2,3]. Este enfoque es problemático cuando se necesita una visión

interior del edificio, o cuando se requiere realizar simulaciones de algún tipo sobre la estructura del edificio.

En este trabajo pretendemos presentar un lenguaje visual que facilite la reconstrucción o el modelado de una estructura arquitectónica de manera más sencilla e intuitiva de lo que es posible en la actualidad, permitiéndonos una libertad absoluta para poder realizar todo tipo de modificaciones. Además, el hecho que este lenguaje esté basado en volúmenes facilita la creación de un interior más realista, al tiempo de proveer de elementos que permitan un estudio más profundo de la estructura.

## 2 LENGUAJE VISUAL BASADO EN VOLÚMENES

Un lenguaje basado en volúmenes es necesario debido a la gran importancia que los elementos arquitectónicos adquieren en las edificaciones del patrimonio arqueológico, donde una pared puede llegar a medir más de un metro de grosor, lo que es fundamental para su estabilidad estructural [4]. Sin embargo, las técnicas de construcción procedural hasta el momento se han centrado siempre en superficies. Muchos de los conceptos utilizados por nosotros están basados en los estudios de Müller et al. [2], pero a diferencia de estos, los nuestros están basados en volúmenes. Por lo tanto, el lenguaje propuesto será la representación visual de un lenguaje para el modelado procedural volumétrico de edificios, formado por reglas con una sintaxis claramente definida.

Este lenguaje visual está formado por nodos, que representan cada una de las operaciones realizadas, y conectados entre sí mostrando así el flujo de la geometría a la hora de recrear un edificio del patrimonio arqueológico.

Cada uno de estos nodos crea u opera sobre los elementos arquitectónicos (por ejemplo, paredes, arcos, columnas...) así como sobre los parámetros de éstos (ancho de la columna). Las operaciones básicas que utilizamos son las siguientes:

**SplitVol:** Esta operación divide el volumen de entrada (llamado predecesor) en dos o más volúmenes de salida (llamados sucesores de la operación) (Figura 1). El usuario puede especificar el número de divisiones y el tamaño relativo (porcentual respecto del total de la estructura) o absoluto de cada una de las mismas.

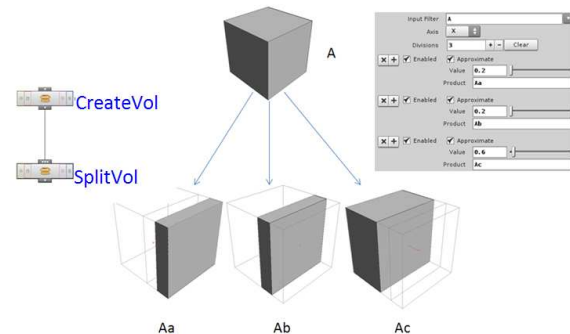


Figura 1. Lenguaje visual, ejemplo e interfaz de la operación SplitVol. En la figura, A es la geometría de entrada y Aa, Ab, Ac las geometrías resultantes.

**RepeatVol:** Esta operación opera sobre un volumen de entrada, dividiéndolo en sub-volúmenes, todos de idéntica medida, tantas veces como sea necesario hasta haber dividido la totalidad del mismo. El usuario sólo debe especificar el tamaño, absoluto o aproximado, de las divisiones (Figura 2).

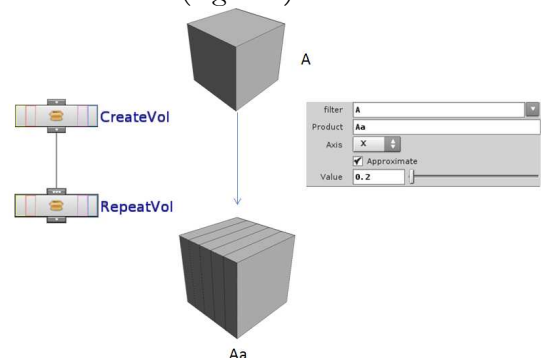


Figura 2. Lenguaje visual, ejemplo e interfaz de la operación RepeatVol.

**ReplaceFaces:** Ésta operación se usa para reemplazar cualquier cara 2D del volumen de entrada por un nuevo volumen que después será procesado por operaciones subsiguientes. Se necesita entrar como parámetros el ID de la cara elegida del volumen de entrada y el espesor del nuevo volumen a crear (Figura 3).

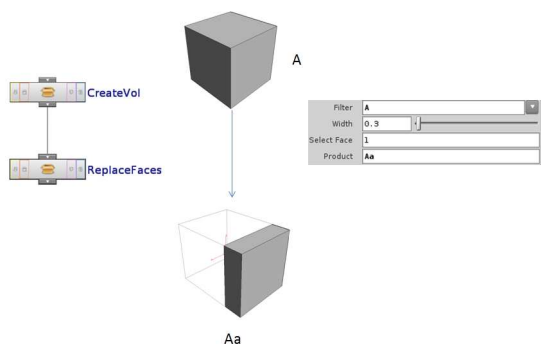


Figura 3. Lenguaje visual, ejemplo e interfaz de la operación ReplaceFaces.

**InsertVol:** Esta operación reemplaza una geometría de entrada por la geometría dada como parámetro. La nueva geometría es escalada y trasladada de manera que la caja englobante de ésta coincida con la de la entrada (Figura 4).

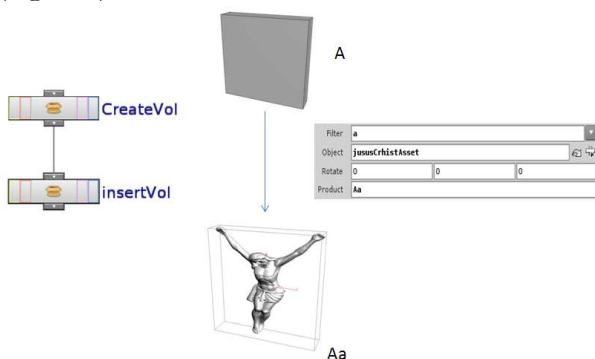


Figura 4. Lenguaje visual, ejemplo e interfaz de la operación InsertVol

**Symmetric:** El objetivo de esta operación es invertir y duplicar la geometría de entrada, produciendo un efecto de “espejo” sobre la geometría. El usuario elige la distancia entre la nueva geometría y la predecesora (Figura 5).

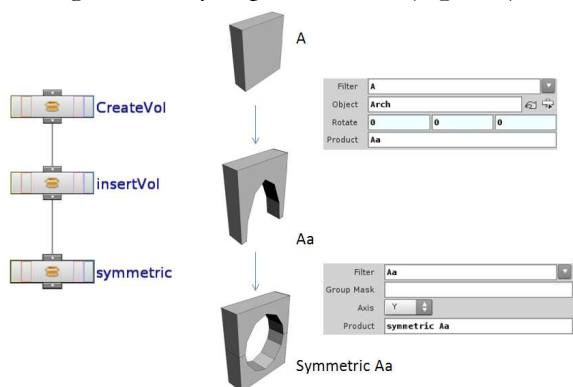


Figura 5. Lenguaje visual, ejemplo e interfaz de la operación Symmetric.

**TransVol:** Esta operación permite escalar, trasladar y rotar la geometría de entrada (Fig. 6).

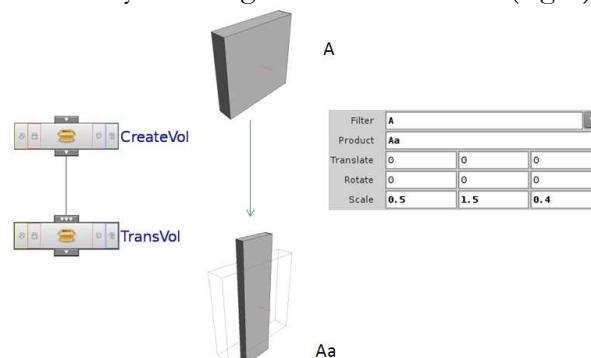


Figura 6. Lenguaje visual, ejemplo e interfaz de la operación TransVol.

Con este lenguaje visual podemos recrear de forma extremadamente realista un edificio o estructura arquitectónica en pocos minutos, al tiempo que nos permite realizar simulaciones físicas para poder así realizar estudios para la conservación de un puente, un acueducto o una iglesia [4].

### 3 CASOS DE USO

**Ejemplo acueducto:** Para la recreación de un acueducto, el usuario decide las dimensiones de éste i el número de plantas que tiene. A cada una de las plantas creamos un arco sostenido por dos columnas y lo repetimos de manera horizontal. De esta manera queda el grafo (Figura 7) y el resultado (Figura 8) siguientes:

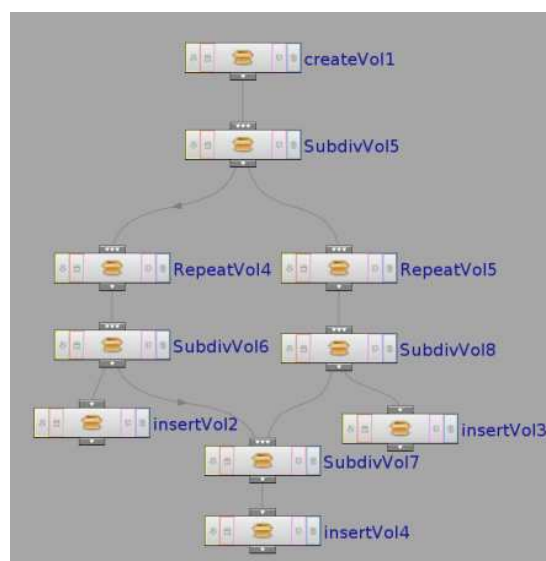


Figura 7. Grafo final del acueducto representado en lenguaje visual.



Figura 8. Resultado final del acueducto representado en un entorno natural.

Dado un volumen inicial con las dimensiones globales del acueducto (Figura 9.A y nodo **CreateVol1** en Figura 7), utilizamos las operaciones descritas en la Sección 2 para lograr el modelo final (Figura 8). A las diferentes plantas creadas (Figura 9.B) con la operación subdivVol (nodo **SubdivVol5** en la Figura 7), les asignamos el ancho de cada uno de los arcos y lo repetimos horizontalmente (Figura 9.C y nodos **RepeatVol4** y **RepeatVol5** de la Figura 7). Descartando la planta más elevada, donde no se construye ningún arco, se sitúan los arcos en dos pasos: En el primero (Figura 9.D), subdividimos verticalmente los elementos en dos partes (nodos **SubdivVol6** y **SubdivVol8** de la Figura 7) e insertamos el arco en la superior con la operación adecuada (nodos **InsertVol2** y **InsertVol3**). En el segundo, justo debajo de cada uno de los arcos volveremos a subdividir (nodo **SubdivVol7**) e insertamos los volúmenes que formaran las columnas del acueducto (Figura 9.E y nodo **InsertVol4**).

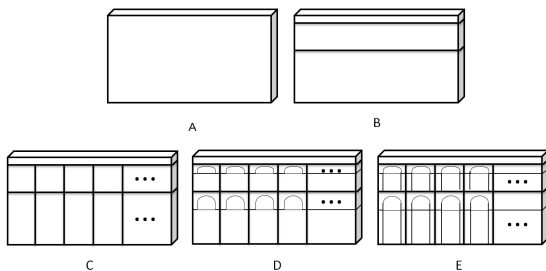


Figura 9. Resultado de aplicar cada una de las operaciones del flujo mostrado en Figura 7

**Ejemplo puente estilo romano:** En este caso (Figura 10) el usuario decide en cuantos pilares se recostará el puente y las medidas correctas tanto de estos como del puente en sí. De esta manera nos queda el grafo (Figura 11). Puede observarse la gran similitud entre los grafos de este ejemplo y del anterior, lo cual muestra el alto grado de re-usabilidad de estos diseños.



Figura 10. Puente acabado en un entorno natural

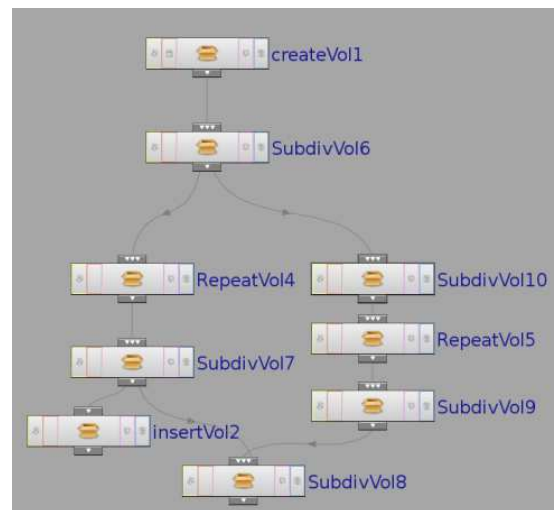


Figura 11. Grafo de operaciones final del puente

**Ejemplo ermita:** Para recrear una pequeña ermita (Figura 13) sólo necesitamos definir el número de ventanas y como estará formada la fachada principal. En nuestro caso, por una ventana circular y una puerta de arco romano. En este ejemplo (Figura 12), hemos agrupado en un solo nodo (nodo **groinVaults2**) los nodos correspondientes a las bóvedas que se encuentran en el interior. Los nodos **windows** y **windows1** también son agrupaciones de los nodos básicos descritos en la Sección 2. Es importante remarcar que, en total, hemos

utilizado menos de 30 nodos para lograr el edificio final.

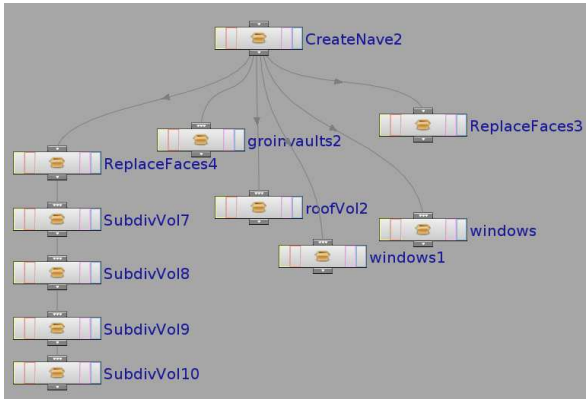


Figura 12. Grafo final de la ermita representado en lenguaje visual.



Figura 13. Resultado final de la ermita

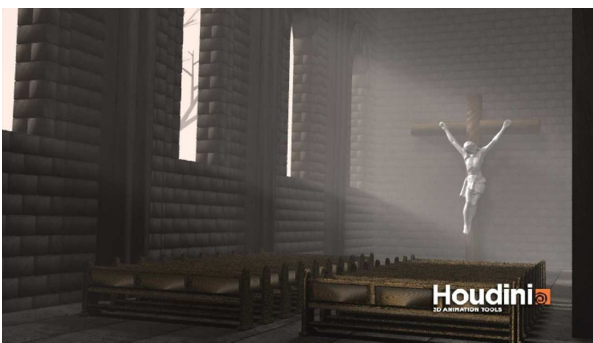


Figura 14. Resultado final del interior de la ermita

En las figuras 13 y 14 podemos observar el resultado de la recreación de la ermita, mientras que en las figuras 15 y 16 observamos la misma ermita, pero añadiendo una degradación estructural a causa del tiempo. Como puede verse, esta degradación es realista gracias al uso de volúmenes, ya que el modelado basado en superficies no permite este tipo de simulaciones.



Figura 15. Resultado final de la ermita añadiendo una degradación estructural.

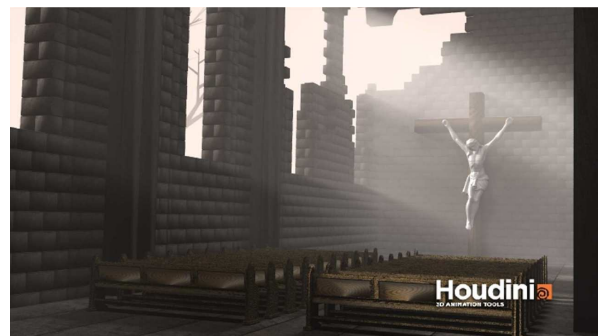


Figura 16. Resultado final del interior de la ermita añadiendo una degradación estructural.

**Ejemplo Iglesia:** En este caso queremos recrear un templo gótico de gran importancia mundial: la Sainte Chapelle, que es un templo gótico situado en Île de la Cité, en el centro de la ciudad de París, Francia. Como puede verse (Figura 17), su estructura apenas ha requerido de unos 42 nodos para quedar correctamente estructurada. La creación requiere, para un usuario no experto, de unos 15 minutos en total para este modelo (Figura 18), y alrededor de un par de horas para un modelo acabado.

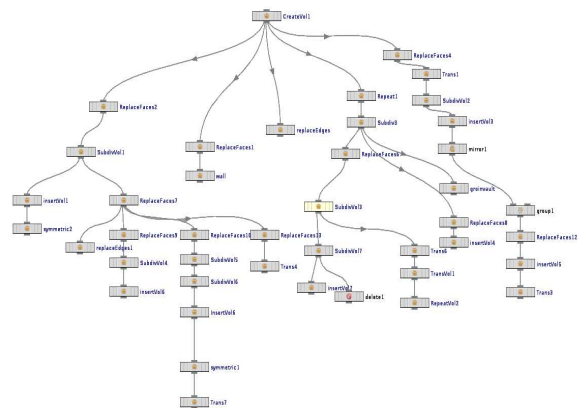


Figura 17. Grafo completo de la iglesia de la figura 18.



Figura 18. Resultado final de la iglesia de la Fig. 17.

## 4 CONCLUSIONES

Como puede apreciarse, el lenguaje propuesto representa una gran ayuda a la hora de generar un prototipo de una estructura arquitectónica compleja, la cual puede completarse en pocos minutos. De la misma manera, un edificio complejo puede completarse en menos de una hora, lo que representa un gran avance en términos de usabilidad respecto del estado del arte actual para este tipo de tareas de modelado.

## AGRADECIMIENTOS

Queremos agradecer a Emily Whiting por los ejemplos que nos facilitó, así como a todos los miembros del Grup de Geometria i Gràfics de la Universitat de Girona por toda su ayuda. Este proyecto fue parcialmente financiado con el proyecto TIN2010-20590-C02-02 del Ministerio de Ciencia e Innovación.

## BIBLIOGRAFÍA

- [1] P. Wonka, M. Wimmer, F. Sillion, and W. Ribarsky. Instant architecture. *ACM Transaction on Graphics*, 22(3):669–677, July 2003. Proceedings ACM SIGGRAPH 2003. 10.
- [2] P. Müller, P. Wonka, S. Haegler A. Ulmer, and L. Van Gool. Procedural modeling of buildings. *ACM Trans. Graph.*, 25 (3):614–623, 2006.
- [3] G. Patow, User-Friendly Graph Editing for Procedural Buildings. *IEEE Computer Graphics and Applications*, vol. 32, no. 2, Mar./Apr. 2012, pp. 66-75; doi:10.1109/MCG.2010.104.
- [4] E. Whiting, J. Ochendorf, F. Durand, Procedural modeling of structurally-sound masonry buildings. *ACMSIGGRAPH Asia 2009 papers* (New York, NY, USA, 2009), SIGGRAPH Asia '09, ACM, pp. 112:1–112.