



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES ACATLÁN

“MODELADO GRÁFICO DE UN CUERPO
NEUMÁTICO CON OpenGL A BASE DE
ECUACIONES DIFERENCIALES”

T E S I S

QUE PARA OBTENER EL GRADO DE:

LICENCIADO EN MATEMÁTICAS
APLICADAS Y COMPUTACIÓN

P R E S E N T A

JORGE ANTONIO GARCÍA GALICIA

DIRECTOR DE TESIS:

DR. MARÍA DEL CARMEN VILLAR PATIÑO

México, D.F.

2008

*A toda la gente que paga sus impuestos
y que cumple con su trabajo en paz.*

Agradecimientos

Agradezco al Consejo Nacional de Ciencia y Tecnología (CONACyT), por apoyar esta investigación con la beca CVU-XXXX para estudios de Maestría en la Universidad Nacional Autónoma de México.

Agradezco también al Departamento de Ciencias de la Computación del Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas por permitirme hacer uso de sus instalaciones y por todos los recursos materiales con los que me apoyaron.

Y por supuesto, gracias...

A toda la gente que paga sus impuestos y que cumple con su trabajo en paz.

Resumen

Este trabajo trata de como visualizar campos escalares que han pasado por un proceso de digitalización. En particular se usa una técnica de graficación por computadora conocida como visualización por superficies o *surface rendering*.

Se asume que se tienen conjuntos digitales de datos que provienen de haber muestreado de manera uniforme el espacio en tres dimensiones. Asumimos que este muestreo está hecho en una rejilla rectangular y por lo tanto tenemos una imagen digital en 3D o volumen. Hacemos dos suposiciones importantes sobre el volumen. Primero, que es una buena aproximación del campo escalar y segundo que no tenemos información de la manera como se realizó la digitalización.

Índice general

Introducción	1
1 Un poco de Matemáticas	3
2 Figuras y otros entornos	5
2.1 Subfiguras	6
3 Ciencias de la computación	9
3.1 Código fuente	10
Conclusiones	11
Bibliografía	13

Índice de figuras

2.1	Distribución normal	5
2.2	Módulo de iluminación de Phong	6

Índice de cuadros

2.1	Tabla Anova para un ejercicio imaginario	7
-----	--	---

Índice de algoritmos

1	Algoritmo de Euclides	9
---	---------------------------------	---

Introducción

La percepción que tenemos los seres humanos del mundo a través de nuestros sentidos es inmediata. Además, dependemos de estos sentidos para realizar la mayoría de las actividades diarias. También hay que señalar que somos criaturas visuales, de nuestros cinco sentidos el que nos proporciona la mayor cantidad de la información que asimilamos sin lugar a dudas son nuestros ojos [3].

El procesamiento de imágenes estudia como procesar imágenes digitales por medios computacionales [2]. Algunas veces este proceso implica transformar una imagen en otra imagen en donde se ha resaltado, o inhibido, cierta característica. Otras veces implica obtener otro tipo de información de una imagen, por ejemplo un histograma. Finalmente, algunas veces implica poder distinguir o interpretar datos de objetos a partir de las imágenes.

Las gráficas por computadora (GC) son la rama de las CC que estudia las técnicas para producir imágenes digitales a partir de descripciones matemáticas. Estas imágenes son visualizadas en dispositivos de despliegue en 2D, por ejemplo un monitor, por lo que generalmente es necesario hacer una proyección de los objetos (en tres dimensiones) a un plano.

La visualización científica se encarga de construir información visual de conjuntos de datos científicos. Estos datos pueden tener orígenes muy diversos; inclusive pueden simular fenómenos de la naturaleza que no pueden ser percibidos por nuestra vista. Por ejemplo, ver en la pantalla el espacio de solución de una ecuación diferencial que modele la vibración de la cuerda de una guitarra es un problema de visualización científica que se podría equiparar con la idea de “visualizar la música”.

En éste trabajo se abarca un poco de estas tres áreas. Asumimos que tenemos un conjunto de datos distribuidos en un arreglo cúbico que representan un muestreo de un

campo escalar tridimensional. Estos datos forman lo que típicamente se conoce como una imagen digital en 3D o volumen. Debido a la naturaleza de los datos nos interesa poder visualizarlos sin perder su información espacial.

Objetivos del trabajo

El principal objetivo de este trabajo es encontrar por medio de la modificación de normales una forma de hacer la superficie obtenida por el Algoritmo de Artzy visualmente mas agradable. Haciendo énfasis en que no hacemos ninguna suposición sobre como fue la discretización del volumen.

Capítulo 1

Un poco de Matemáticas

Sólo para probar el siguiente texto está en ingles: In god we thrust.

En éste capítulo se presentan varios ejemplos com muchas matemáticas. Primero, se puede crear una ecuación simple de la siguiente manera:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \quad (1.1)$$

Y podemos hace una referencia a ella en el texto de la siguiente manera: La ecuación (1.1) representa una elipse. Si no queremos que las ecuaciones esten numeradas podemos hacer:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

También se pueden insertar ecuaciones dentro de un párrafo, por ejemplo: $\forall x \in \mathbb{R}$. Se pueden poner links a un sitio web de la siguiente manera: Para aprender acerca de integrales y sumatorias puedes leer el siguiente [wikilibro](#) o puedes buscarlo en [www.google.com](#). Nótese que la segunda forma cambia la fuente del texto.

Éste es un ejemplo de una función con casos, como si fuera una *pdf*.

$$f(y) = \begin{cases} \frac{1}{25}y & \text{si } 0 \leq y < 5 \\ \frac{2}{25} - \frac{1}{25}y & \text{si } 5 \leq y < 10 \\ 0 & \text{si } y < 0 \text{ ó } y > 10 \end{cases} \quad (1.2)$$

Éste es un ejemplo de paréntesis que ajustan su tamaño automáticamente:

$$P\left(A=2\left|\frac{A^2}{B}>4\right.\right)$$

Finalmente, pongo un ejemplo de como escribir una serie de pasos matemáticos usando el entorno: **align**. Poner * dentro del entorno te permite omitir los números

$$\begin{aligned}P(X \leq 3) &= \int_0^3 \frac{1}{25}y \, dy \\&= \frac{1}{25} \cdot \frac{1}{2}y^2 \Big|_0^3 \\&= \frac{1}{25} \left(\frac{1}{2}9 - \frac{1}{2}0 \right) = \frac{1}{25} \cdot \frac{9}{2} = \frac{9}{50} \approx 0.18\end{aligned}$$

Capítulo 2

Fíguas y otros entornos

En éste capítulo hay ejemplos de como poner figuras. La manera mas común es que las figuras aparezcan centrados junto con su pie de figura.

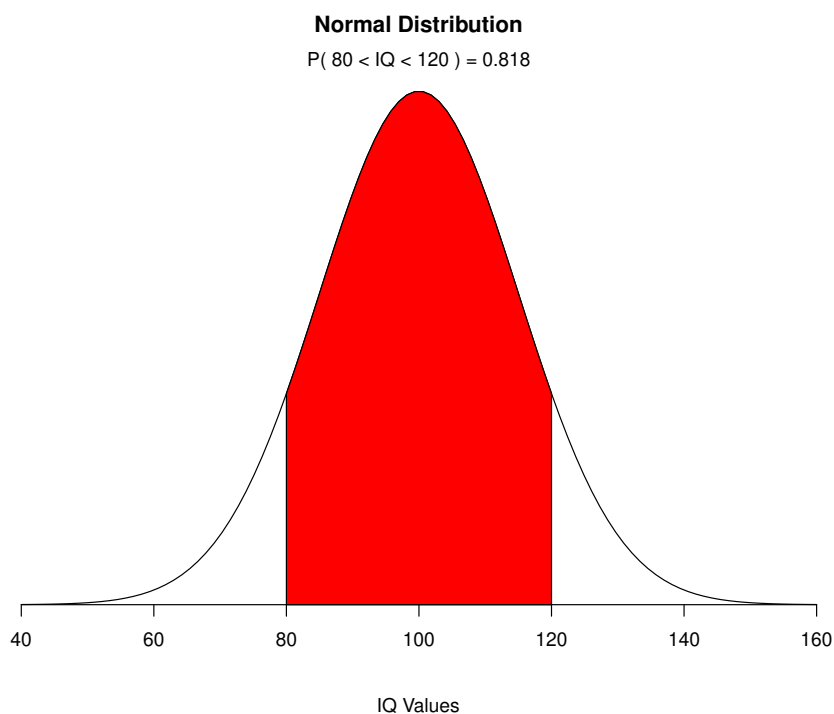


Figura 2.1: Gráfica de una distribución normal. Fue creado usando el siguiente [script en R](#).

2.1 Subfiguras

También es posible poner figuras, compuestas de varias subfiguras. Cada subfigura tiene su propio pié y hay un pié de figura extra para todo el grupo. Posteriormente, es posible referirte a toda la figura así: Vease la Figura 2.2 ó referirte a una subfigura así: Ve la Sub Figura 2.2b.

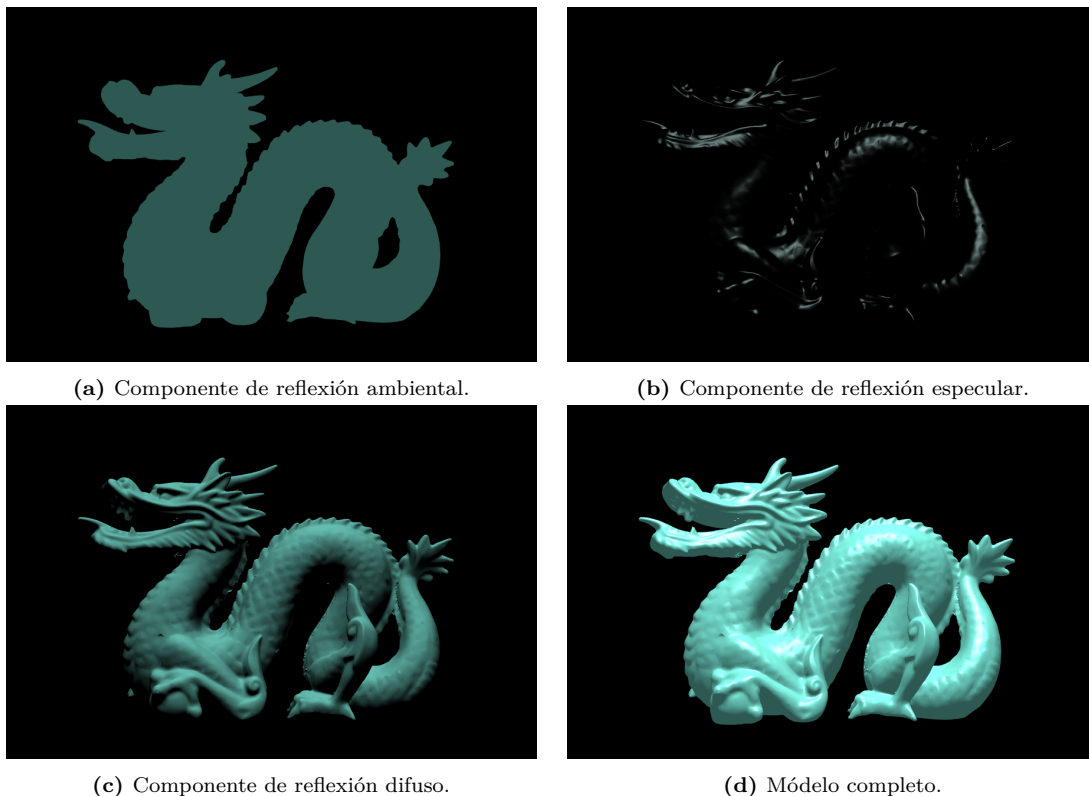


Figura 2.2: Componentes del modelo de iluminación de Phong.

Así es como se cita un libro: éste ejemplo fué tomado de [2]. También hay un ejemplo de como hacer que un libro aparezca en las referencias sin que este citado explicitamnet en el texto. Por último, este es un ejemplo de una tabla muy elegante: Table 2.1

Source	DF	SS	MS	F	P-value
Modelo	2	0.00318564	0.00159282	7.72	0.0014
Error	42	0.00866760	0.00020637		
Total	44	0.01185324			

Cuadro 2.1: Tabla Anova para un ejercicio imaginario

Capítulo 3

Ciencias de la computación

En éste capítulo, dare algunos tips enfocados a ls ciencias de la computación

Primero voy a mostrar como incluir algoritmos en forma de pseudocódigo. Y desde luego se incluyen en el índice y se pueden referenciar así: El Algoritmo 1 es el primer algoritmo en la historia. Se puede referenciar una línea del algoritmo. El ciclo while termina en la línea line 7. Las ideas las tomé de este enlace [wikibook](#) y de éste [post](#).

Por defecto el entorno de algoritmo usa todo el ancho de la página. Es decir, se sale de los márgenes. Hay un [truco](#) para hacerlo entrar en un cierto ancho. Sin embargo, recomiendo usar el truco con moderación.

Algoritmo 1 Algoritmo de Euclides

```
1: procedure EUCLID( $a, b$ ) ▷ El g.c.d. de  $a$  y  $b$ 
2:    $r \leftarrow a \bmod b$ 
3:   while  $r \neq 0$  do ▷ Si  $r = 0$  ya tenemos la respuesta
4:      $a \leftarrow b$ 
5:      $b \leftarrow r$ 
6:      $r \leftarrow a \bmod b$ 
7:   end while
8:   return  $b$  ▷  $gcd = b$ 
9: end procedure
```

```
int main() {  
    printf("hello, world");  
    return 0;  
}
```

Listado 3.1: Un programa de ejemplo en C

```
void insertion_sort(array<int, N>& input) {  
    for (size_t i = 0; i < N; ++i) {  
        size_t max_key = 0;  
        for (size_t j = 1; j < N - i; ++j) {  
            if (input[j] > input[max_key]) {  
                max_key = j;  
            }  
        }  
        if (max_key != N - i - 1) {  
            auto tmp = input[N - i - 1];  
            input[N - i - 1] = input[max_key];  
            input[max_key] = tmp;  
        }  
    }  
}
```

Listado 3.2: Una implementación defectuosa de insertion sort

3.1 Código fuente

Aquí se muestra como incluir código fuente usando el paquete `mined`. Este es un ejemplo en el lenguaje C.

Este es otro ejemplo de como incluir Python dentro de un parrafo: `print(x**2)`. Finalmente, lo mas util es incluir el codigo fuente desde un archivo externo: Vean el Listado 3.2 como ejemplo. Me ayude muchisimo de [aquí](#) y de la [ayuda de Overleaf](#). Estamos usando el mismo hack que usamos con los algoritmos para hacer el listado entrar dentro de los margenes de la página.

Conclusiones

Hay que señalar que los resultados son puramente visuales y altamente dependientes del modelo de iluminación. Aunque en este trabajo nos limitamos al modelo de iluminación de Phong y al sombreado de Gouraud, bien valdría la pena explorar modelos alternativos tanto para iluminación como para sombreado.

También como trabajo futuro se podría intentar que el GPU se hiciera cargo no solo de los cálculos de iluminación si no también de los cálculos de extracción de superficie, es decir que el Algoritmo de Artzy debería de poderse implementar en GPU.

La elección del blob Kaiser-Bessel para la construcción de la superficie implícita fue en gran medida al conocimiento previo del grupo de trabajo y a que ha dado buenos resultados para fines de visualización. Sin embargo, este blob es relativamente complejo de evaluar. Por esto no se descarta la posibilidad de usar algún otro blob y que pueda dar resultados parecidos y con mucho menos costo computacional.

Tradicionalmente se han usado técnicas de procesamiento de imágenes para mejorar la visualización de imágenes en biomedicina. Los resultados de éste trabajo demuestran que es posible mejorar la visualización de una imagen 3D usando técnicas de iluminación. Por lo tanto, debe de seguirse investigando como mejorar las imágenes no solo con procesamiento de imágenes, si no también por técnicas de graficación por computadora.

Bibliografía

- [1] Edgar Garduño, Gabor T. Herman y H. Katz. «Boundary tracking in 3D binary images to produce rhombic faces for a dodecahedral model». En: *IEEE Transactions on Medical Imaging* 17.6 (1998), págs. 1097-1100.
- [2] Rafael C. Gonzalez y Richard E. Woods. *Digital Image Processing*. Inglés. 2ª ed. Prentice Hall, 2002.
- [3] Hans Hagen y col. «Scientific Visualization: Methods and Applications». En: *Proceedings of the 19th spring conference on Computer graphics*. Budmerice, Slovakia: ACM, 2003, págs. 23-33.
- [4] Robert M. Lewitt. «Multidimensional digital image representations using generalized Kaiser-Bessel window functions». En: *Journal of the Optical Society of America* 7.10 (1990), págs. 1834-1846.
- [5] Bui Tuong Phong. «Illumination of Computer-Generated Images». Tesis doct. University of Utah, 1973.
- [6] The Khronos Group. *The Industry Standard for High Performance Graphics*. <http://www.opengl.org/>. Mayo de 2011.
- [7] Wikipedia, the free encyclopedia. *Anatomical terms of location*. http://en.wikipedia.org/wiki/Anatomical_terms_of_location. Mayo de 2011.