

# Trabajo Práctico 1

## “Soda Stereo”

Francisco Gómez Fernández (Pachi)

Métodos Numéricos  
Departamento de Computación  
Facultad de Ciencias Exactas y Naturales  
Universidad de Buenos Aires

18 de agosto de 2017



**DEPARTAMENTO  
DE COMPUTACION**

Facultad de Ciencias Exactas y Naturales - UBA

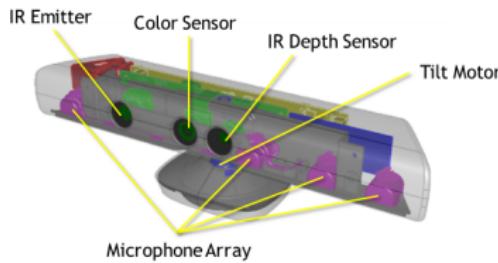
# Motivación

El proceso de digitalización 3D consiste en conseguir una representación o modelo computacional de los objetos del mundo real que nos rodean. Si bien hace muchos años que existen los sensores 3D, en la actualidad están volviendo a cobrar importancia gracias al creciente auge de las impresoras 3D.

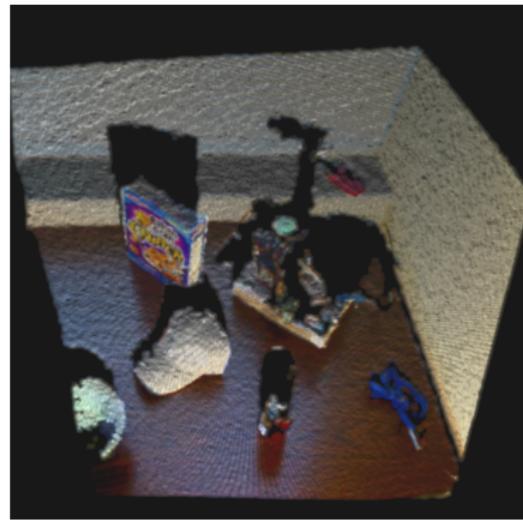
Existen muchos tipos de sensores 3D que nos permiten obtener la geometría y/o apariencia de escenas tridimensionales, como los LiDAR basados en luz láser pulsada, la popular Kinect basada en técnicas de luz estructurada y en su nueva versión usando cámaras de tipo *time-of-flight*, cámaras estéreo, entre otros.

# Motivación

Sensores RGBD (Color más profundidad)



Sensor Microsoft Kinect.



Vista 3D generada con una captura de Kinect.

Para más información, ver el curso dictado en 2013 por Gabriel Taubin en el DC.  
“Introducción a la fotografía 3D” <http://mesh.brown.edu/3DP-FCEN-2013/pdfs/3DP-FCEN-2013-01.pdf>

# Fotometría estéreo

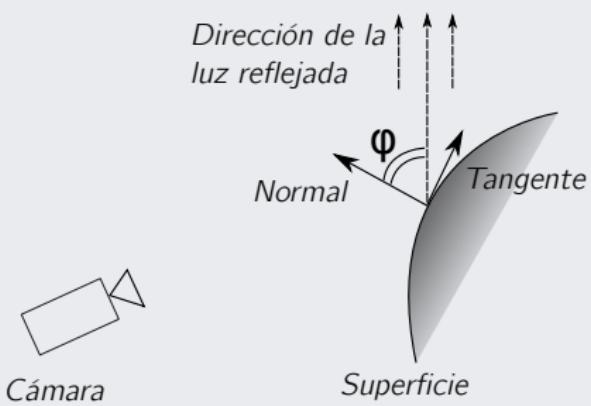
Cuando miramos una fotografía o vemos la televisión, no vemos una imagen en 2D, sino que entendemos lo que vemos como un objeto en 3 dimensiones. Es decir, podemos inferir la forma 3D de los objetos a partir de imágenes 2D. La pregunta es: podemos simular esta habilidad con una computadora?

La naturaleza de la iluminación y su ubicación pueden cambiar drásticamente la apariencia de una escena. Para poder reconstruir la forma de un objeto 3D a partir de una imagen 2D plana primero debemos entender el proceso de formación de las imágenes.

# Modelo de creación de las imágenes

El modelo más simple que describe la formación de una imagen supone que la cámara está lo suficientemente lejos de la escena como para tener una representación escalada del objeto completo, la iluminación es puntual y está suficientemente alejada, por ejemplo el sol mismo. Por último, se considera que la superficie del objeto es esencialmente mate y refleja la luz uniformemente en todas direcciones. Esta clase de superficies se denominan Lambertianas.

El brillo de un objeto de superficie Lambertiana depende de la orientación de la superficie con respecto a la fuente de luz. La intensidad de luz incidente se denota  $I_0$  y el ángulo entre la luz incidente y el vector normal a la superficie se denomina  $\phi$ .



# Creación de una imagen

La intensidad registrada en cada punto  $(x, y)$  de la imagen está dada por:

$$I(x, y) = I_0 \rho \cos(\phi) = I_0 \rho (n_x s_x + n_y s_y + n_z s_z)$$

donde la normal  $\mathbf{n}$ , y la dirección de la fuente de luz  $\mathbf{s}$ , están dadas por:

$$\mathbf{n} = (n_x, n_y, n_z)^t \quad \mathbf{s} = (s_x, s_y, s_z)^t$$

La variable  $\rho$  es un parámetro que representa las propiedades reflectivas (albedo) de la superficie en la posición  $(x, y)$ . Debido a que consideramos que la superficie es Lambertiana, la intensidad registrada no depende de la posición de la cámara. Los valores de gris de la imagen captados por la cámara dependen esencialmente de la orientación de la superficie con respecto a la fuente de luz. La variable  $\rho$  es un parámetro que representa las propiedades reflectivas (albedo) de la superficie en la posición  $(x, y)$ . Debido a que consideramos que la superficie es Lambertiana, la intensidad registrada no depende de la posición de la cámara. Los valores de gris de la imagen captados por la cámara dependen esencialmente de la orientación de la superficie con respecto a la fuente de luz.

# Método de reconstrucción 3D

El objetivo de la aplicación de reconstrucción 3D es encontrar el modelo 3D de un objeto quieto a partir de tres imágenes<sup>1</sup> del mismo con distinta iluminación. El modelo 3D son los puntos  $(x, y, z)$  y sus normales.

## Calibración (única vez)

Tomando de modelo una esfera iluminada bajo las mismas condiciones que los objetos que se quiere modelar, y como la geometría en 3D de la esfera es conocida, de aquí se puede inferir las direcciones de iluminación.

## Reconstrucción 3D (por cada objeto)

Una vez encontradas estas direcciones se aplican para resolver el problema para cada objeto:

- ① Estimación de las normales  $\mathbf{n}$
- ② Calculo de las profundidades  $z$

<sup>1</sup>En general se capturan más de tres imágenes, en este caso se deben elegir las mejores 3 con algún criterio.

# Calibración

Cálculo de las direcciones de iluminación  $s$ .

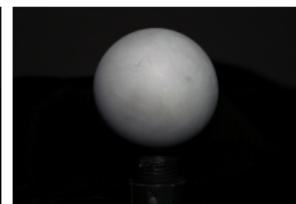
La secuencia de imágenes de la esfera (mate) nos servirá para calibrar el sistema por **única vez** y así obtener las direcciones de iluminación en cada caso  $s^1, s^2, \dots, s^k$ , con  $k = 12$  para nuestro conjunto imágenes.



$s^1$



$s^3$



$s^6$



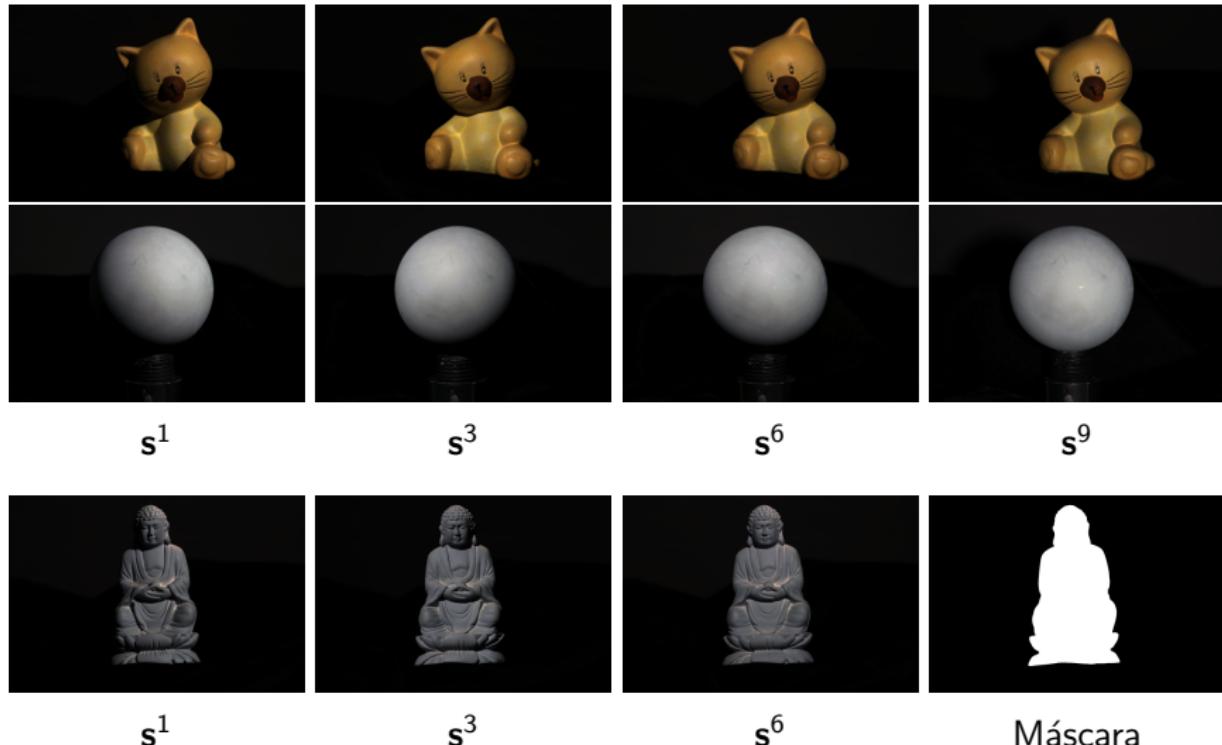
$s^9$

Por lo tanto, el punto más brillante sobre la imagen de la esfera será el que corresponderá a la dirección de iluminación buscada y en dirección opuesta al normal ese punto **Sugerencia:** este punto puede encontrarse buscando en la imagen el píxel de máximo valor en un entorno..

Comparar las direcciones obtenidas por las provistas por la cátedra en luces.txt.

# Conjunto de imágenes de objetos

Secuencia de 12 imágenes por objeto más una máscara que indica los píxeles válidos.  
Para cada objeto, la imagen  $i$ -ésima se corresponde con la iluminación  $i$ -ésima.

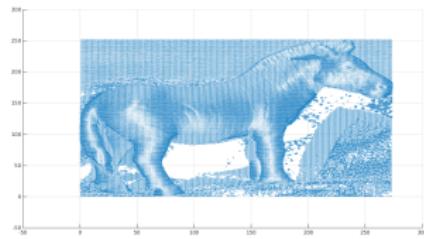


# Reconstrucción 3D (paso 1)

## Estimación de las normales

Aquí se deben elegir 3 direcciones de iluminación de todas las obtenidas en la calibración y resolver el siguiente sistema para cada píxel, y luego despejar cada  $\mathbf{n}$ .

$$\begin{pmatrix} s_x^1 & s_y^1 & s_z^1 \\ s_x^2 & s_y^2 & s_z^2 \\ s_x^3 & s_y^3 & s_z^3 \end{pmatrix} \begin{pmatrix} m_x \\ m_y \\ m_z \end{pmatrix} = \begin{pmatrix} l_1 \\ l_2 \\ l_3 \end{pmatrix}$$

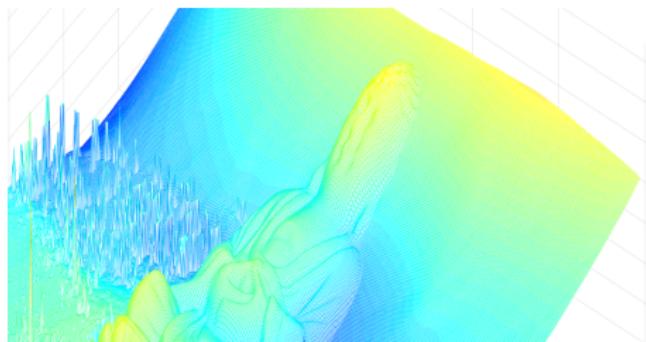
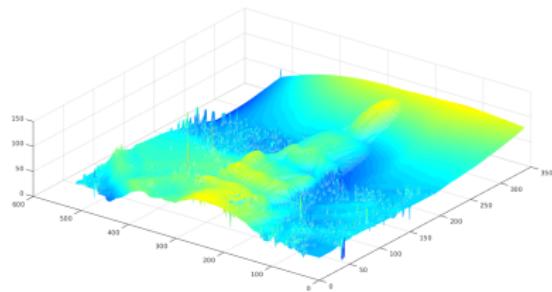


Notar que la elección de las 3 direcciones tiene gran impacto en el cálculo de las normales y pueden no ser las mismas para diferentes objetos.

# Reconstrucción 3D (paso 2)

## Estimación de las profundidades

Por último debe plantearse y resolverse el sistema de ecuaciones lineales para encontrar cada  $z$ . El sistema a resolver cuya matriz es  $A = M^t M$ , tiene dimensión  $2N \times 2N$  donde  $N$  de la cantidad de píxeles de la imagen.



Más ejemplos y resultados en:

[http://www-2.dc.uba.ar/materias/metnum/dnload/2017\\_C2/tp1/resultados\\_y\\_ejemplos.zip](http://www-2.dc.uba.ar/materias/metnum/dnload/2017_C2/tp1/resultados_y_ejemplos.zip)

## Sugerencias de visualización

- Utilizar los resultados provistos para comparar con los propios.
- Para leer matrices y visualizar en Matlab, primero exportarlas desde C++ en texto plano.
- Utilizar `quiver3` para dibujar normales y `surf` y `mesh` para los modelos completos.
- Es posible exportar las matrices de profundidad/normales como imágenes PGM/PPM para una rápida visualización perdiendo precisión. Para realizar esto, llevar todos los valores al rango [0, 1], multiplicar por 255 y luego convertir a `uchar`. Ver script `map2rgb.m`

## Ejemplos de visualización en:

[http://www-2.dc.uba.ar/materias/metnum/dnload/2017\\_C2/tp1/resultados\\_y\\_ejemplos.zip](http://www-2.dc.uba.ar/materias/metnum/dnload/2017_C2/tp1/resultados_y_ejemplos.zip)

# Sugerencia de avance para el TP

## Tareas por deadline

- 23 de agosto:
  - Leer enunciado y pautas de laboratorio. Descargar material del TP.
  - Código: operaciones matriciales + I/O de imágenes
  - Informe: sección introducción
- 25 de agosto:
  - Código: estimación de normales
- 1 de septiembre:
  - Código: estimación de profundidades y calibración
  - Informe: sección desarrollo
- 8 de septiembre
  - Informe: sección resultados y discusión
  - Código: comentar y preparar .zip para la entrega

## Atención

En caso de trabarse en alguna etapa, utilizar los datos intermedios provistos por la cátedra en el archivo:

[http://www-2.dc.uba.ar/materias/metnum/dnload/2017\\_C2/tp1/resultados\\_y\\_ejemplos.zip](http://www-2.dc.uba.ar/materias/metnum/dnload/2017_C2/tp1/resultados_y_ejemplos.zip)

# Y para los inquietos...

## Puntos opcionales pero interesantes

- Calcular 3 direcciones de iluminación automáticamente.
- Se podría calcular las máscaras de los objetos automáticamente?
- Pesar las normales por su intensidad para tener una mejor estimación de  $z$
- Replicar el experimento en sus casas usando una cámara, un trípode y una lámpara :oO