

Escaner Laser 3D

Quincho Mamani, Lehi lquincho@unsa.edu.pe

Escuela Profesional de Ciencia de la Computación - Universidad Nacional de San Agustín (UNSA)

Arequipa, Peru

Resumen—Un escáner 3D es un dispositivo que digitaliza un objeto real para poder ser usado en animaciones 3D, para ser impreso (usando una impresora 3D) para hacer simulaciones, entre otras aplicaciones. La finalidad de este proyecto es construir un escáner 3D usando un láser que nos ayudará a poder obtener la silueta de un objeto a partir de líneas para luego ser convertido en una línea de puntos en 2 dimensiones por cada fotograma y finalmente unir todas las líneas usando transformaciones y obtener una nube de puntos en 3 dimensiones.

Index Terms—Compilador, Lenguaje de programación, C, Analizador sintáctico

I. MARCO TEORICO

II. RECURSOS

II-A. Hardware

Como parte del Hardware de nuevo escaner estamos usando los siguientes elementos:

- Cámara web
- Caja (el tamaño depende del tamaño de los objetos que se quiera escanear).
- Plataforma giratoria
- Motor de 5v.
- Potenciómetro
- Láser de 5v.
- Cables
- Batería de 5v de salida
- Una computadora o laptop
- Led pequeño de cristal

II-B. Software

El código fue hecho en C++, con las siguientes librerías:

- OpenCV
- PCL (Point Cloud Library)
- VTK (Visualization Took Kit)

III. CONSTRUCCIÓN DE LA PLATAFORMA

Para la construcción de la plataforma se tiene que tener en cuenta muchos factores que pueden afectar al resultado final, los más importantes son los siguientes:

- La posición de la camara y el laser
- La velocidad a la que gira la plataforma
- La luz

III-A. Láser y Cámara

Los láser generalmente solo nos generan un punto de luz cuando el haz de luz entra en contacto con una superficie. Para esto usamos un diodo led para generar una línea a partir del punto del láser. La idea principal del láser es obtener una línea de luz sobre la superficie del objeto que estará en constante rotación, de tal manera que esta línea vaya tomando diferentes formas. Todas estas líneas unidas de forma cilíndrica nos ayudaran a obtener una representación del objeto a partir de líneas que pueden que transformaremos en una nube de puntos.



Figura 1. Luz laser del escáner

La cámara debe enfocar la plataforma justo en el medio, por esto dibujamos una línea de referencia en el gráfico obtenido por la cámara para poder tener la ubicación correcta de la cámara.

El ángulo que forme la posición de la cámara con el láser que generará la línea no debe ser muy grande por que la cámara no podrá captar la línea de luz, y mucha información se perderá, tampoco debe ser muy pequeño ya que mientras más pequeño es el ángulo, la línea es cada vez más recta, si el ángulo fuera de 0° la línea siempre sería recta, y al unir las líneas siempre obtendríamos un cilindro. Mientras más alejemos la línea el objeto tomará una mejor forma. Por tanto el ángulo que se consideró adecuado para no perder información y tener una mejor representación de la silueta del objeto real fue de 45°

La velocidad a la que gira la plataforma necesita ser regulada para poder calibrar la velocidad y el tiempo que se

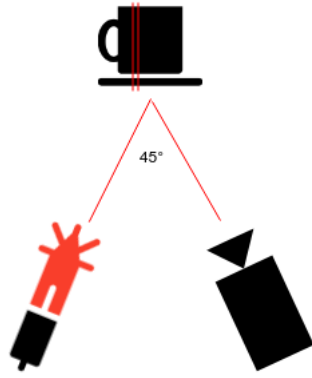


Figura 2. Caption

capturan las líneas. Se conectó al motor un potenciómetro para controlar la velocidad de la plataforma.

La luz sobre el objeto es también importante, si hay mucha luz no se podrá reconocer solo la línea que deseamos con facilidad. Por ese motivo se usó una caja.

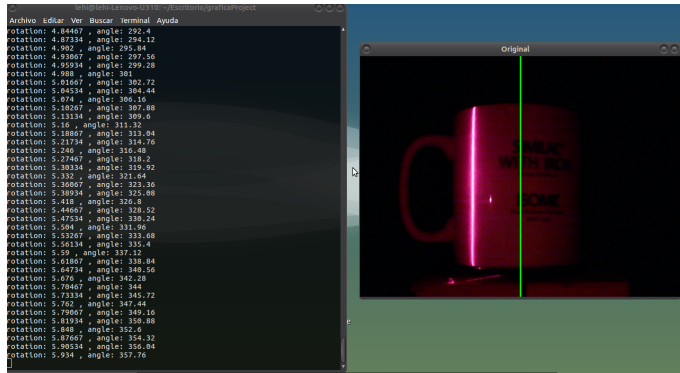


Figura 3. Capturando línea que genera la luz laser

IV. PROCESAMIENTO DE FOTOGRAMAS

Luego de obtener la línea de luz sobre el objeto en rotación sobre su eje central usamos OpenCV para usar la cámara web y aplicar algunos filtros y reducir el brillo de cada fotograma. Usamos las siguientes funciones y pasos:

```
void cvtColor(frame, edges, CV_BGR2HSV);
edges=edges + Scalar(1, 10, 10);
void inRange(...);
void erode(...);
void resize(...);
void Canny();
```

cvtColor nos sirve para convertir la matriz de píxeles a los 3 canales HSV (Hue, Saturation, Value).

edges=edges + Scalar(1, 10, 10); sirve para cambiar el brillo de la imagen, dependiendo de la iluminación del escenario es que es necesario subir o bajar el brillo para que la línea del láser pueda verse mejor.

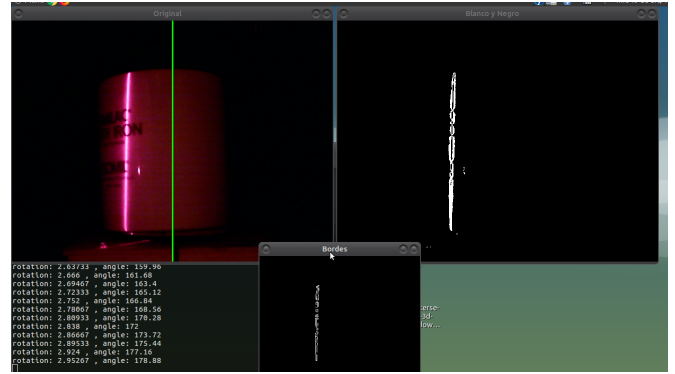


Figura 4. Procesamiento de fotografías

`void inRange(...)`; sirve para binarizar la imagen dependiendo de los umbrales de cada canal HSV. Esto nos ayudará a solo visualizar lo que más brille de la imagen, que es la línea generada por el láser.

Usamos `void erode(...)`; para adelgazar la línea que obtengamos después de binarizar la imagen.

La función `void resize(...)`; se usó para reducir el tamaño de la imagen para reducir el cálculo de puntos.

La función `Canny()`; Nos ayuda a solo obtener los bordes para hacer menos cálculo y tener un resultado más preciso.

El resultado de las funciones anteriores es el mostrado en las imágenes.

V. CALCULO DE DISTANCIAS

Teniendo como base el frame segmentado. Tomaremos cada punto en Y y aproximaremos la coordenada en X de acuerdo a la distancia entre el haz de luz y la línea de dirección de la cámara.

Esta distancia será obtenida en píxeles. Se pretende escalar esta distancia con la equivalencia: 100px = 1,0. La coordenada Y toma la misma escala. Para conseguir la coordenada Z, que representa la profundidad, de acuerdo al esquema es necesario hallar la distancia r, desde el punto actual al eje y que coincide con el centro del cuerpo. Usando trigonometría, y teniendo en cuenta que conocemos x, luego de la aproximación visual, y el ángulo de rotación de acuerdo a la disposición entre la cámara y el láser.

Aplicamos la siguiente fórmula de coordenadas polares:

$$x = r \cdot \sin(\Theta) \quad (1)$$

Después de haber obtenido la distancia r, necesitamos calcular la profundidad usando pitágoras

$$r^2 = x^2 + Z^2 \quad (2)$$

Una vez obtenidos las 3 coordenadas de un punto, necesitamos seguir el mismo proceso con todos los puntos de cada línea del láser.

Para obtener la rotación en los 360° necesitaremos aplicar una rotación con respecto al eje Y, para rotar las coordenadas

X y Y en un ángulo de rotación usando la formula de coordenadas polares .

$$x' = x.\cos(rotacion) \quad (3)$$

$$z' = x.\sin(rotacion) \quad (4)$$

Y con esto ya tenemos los puntos alrededor del eje que formarán la nube de puntos.

VI. GENERACIÓN DE LA NUBE DE PUNTOS

Una vez ya obtenidos los puntos del objeto que escaneamos, usamos el formato de la librería PCL para luego usar sus funciones y poder visualizarlos. La extensión de este formato es .pcd y almacena los puntos en 3 columnas x y z. La siguiente imagen es el resultado de la visualización usando las funciones de la librería PCL.

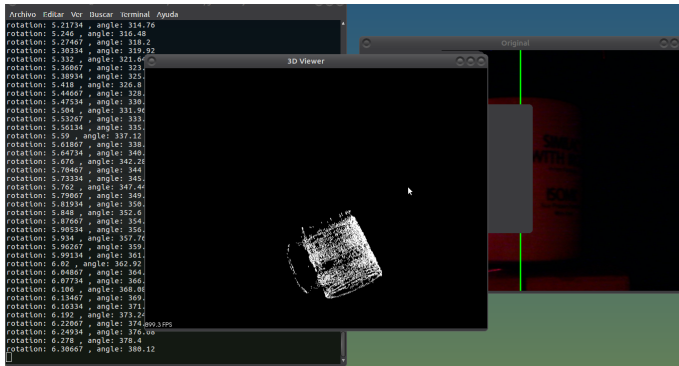


Figura 5. Nube de Puntos obtenido por la union de las lineas de luz

VII. CONCLUSIONES

La implementación del escáner láser es relativamente sencilla, y requiere de materiales fáciles de conseguir, pero también existen muchas desventajas que tiene el escáner láser en comparación con otros métodos como depende mucho de la iluminación y la intensidad del láser que se tenga, y del objeto que se vaya a escanear, si el objeto, si el objeto tiene partes de color negro la luz del láser no brillará tanto en esa parte de la superficie, por tanto no se podrán obtener esos puntos y nuestro objeto tendrá agujeros. Este escáner se podría mejorar si se usan dos punteros láser de ambos extremos con el mismo ángulo de separación con la cámara para cubrir los agujeros que se generen con el otro láser.

El código fuente se encuentra en <https://github.com/lehi10/Computaci-n-Grafica/tree/master/Proyecto%20Escaner%20Laser%203D>

VIII. RESULTADOS



Figura 6. Silla

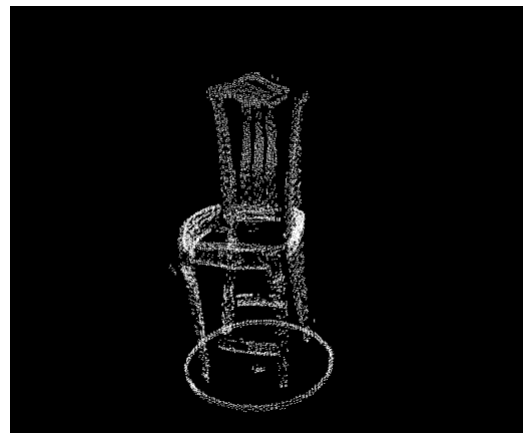


Figura 7. Nube de puntos Silla



Figura 8. Cabeza de Muñeca

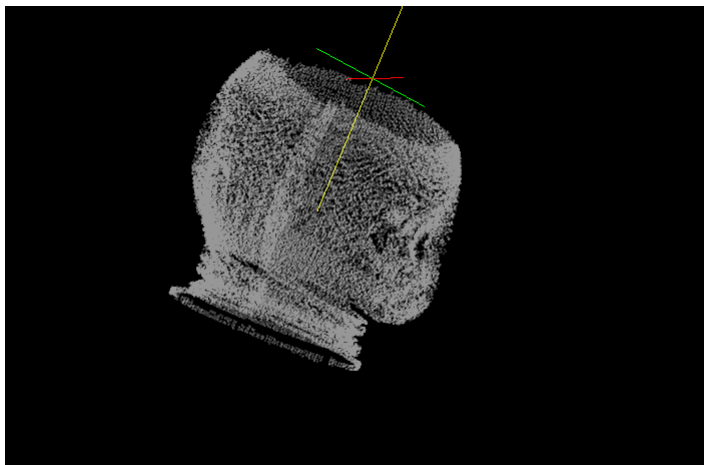


Figura 9. Nube de puntos Cabeza



Figura 10. Erizo

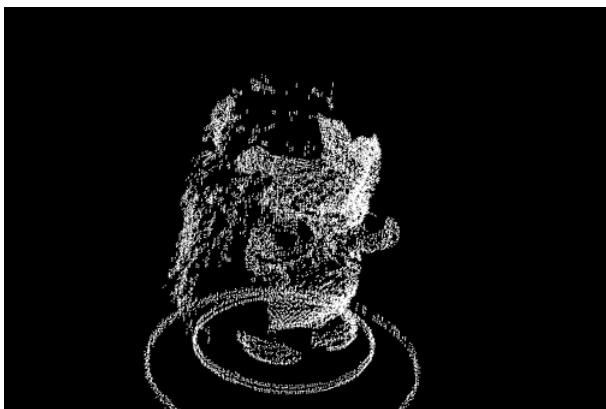


Figura 11. Nube de puntos Erizo