



Plan de Trabajo Final

Carrera Ingeniería de Sistemas

Facultad de Ciencias Exactas – UNICEN

Reconstrucción de objetos en tres dimensiones utilizando Kinect.

Tema: Aplicación para poder crear modelos 3d de objetos escaneados mediante la utilización del sensor Kinect de Microsoft.

Alumno: Chierichetti Julián.

Director: Dr. Juan Pablo D´Amato

Codirector: Dr. Cristian García Bauza

1 Introducción

La nueva tecnología introducida por el dispositivo Microsoft Kinect [1], permite extraer información visual (utilizando una cámara RGB), e información de distancia de cada uno de los objetos que aparecen en escena utilizando técnicas de luz estructurada y almacenarlo como un mapa de profundidades. Esta información puede luego ser proyectada como una nube de puntos en el espacio tridimensional y a continuación convertida en una geometría o malla utilizando los elementos cercanos a cada pixel para finalmente pueda ser visualizada en tiempo real.

Para crear una representación 3D completa de un modelo que luego se pueda utilizar en juegos, aplicaciones de simulación física o diseño gráfico e industrial, se requieren varias capturas desde diferentes puntos de vista del objeto real y fusionarlos. Estas representaciones generalmente presentarán huecos donde no se ha podido capturar datos, tendrán ruido, y a su vez deben ser luego combinadas en una única representación [12]. Si se propone capturar un solo objeto de interés, se debe eliminar la información redundante y la que proviene del resto de la escena de forma coherente. El resultado es una malla con una superficie irregular, producto del error en la captura. Para mejorar el aspecto de la misma, suelen aplicar técnicas de suavizado en el espacio, como puede ser Poisson [13]

Existen diversas estrategias, que pueden ser utilizadas juntas para resolver diferentes cuestiones de estos problemas. Una estrategia es la de *súper-resolución* [2], que plantea tomar múltiples imágenes desde un mismo plano, y mediante un sistema de interpolación, reducir el ruido de la geometría. Para resolver el problema de “huecos” (los datos no capturados), se pueden aplicar estrategias de *estimación* en función de la forma de la malla [3]. Para resolver el problema de combinar múltiples capturas en una única imagen resultado, se utiliza



comúnmente el proceso denominado *registración*, en el que se debe ingresar información al sistema del método de captura, tal como el punto de vista y la orientación del objeto para luego unificar la información utilizando transformaciones en el espacio [4]. Para eliminar el resto de los objetos de la escena y tratar un único objeto por vez, es viable utilizar un filtro de distancia y técnicas de sustracción de *fondo*.

2 Motivación

El sensor Kinect de Microsoft tiene un potencial enorme para poder desarrollar diversas aplicaciones entre ellas, las más conocidas están relacionadas con interfaz natural, pero como se explicó anteriormente el mismo mapa de profundidad que permite detectar una persona puede ser utilizado en la manera correcta para escanear objetos 3D. Al ser un producto relativamente nuevo no está muy desarrollado la utilización del Kinect como escáner 3D, por lo tanto es un desafío poder utilizar nuevas tecnologías y combinarlas con los conocimientos adquiridos en procesamiento de imagen y reconstrucción de mallas tridimensionales con el fin de lograr automatizar proceso que hoy en día llevaría un costo elevado en productos y tiempo. Existen algunos softwares privativos que trabajan en este campo, entre ellos el más conocido Kinect Fusion [6] [9] [10] aunque trabaja con la reconstrucción tridimensional utilizando el Kinect no se especializa en obtener un objeto solo, para ellos existen otras alternativas como KScan3d [7] y Artec3d [8] entre otras, pero la propuesta es realizar un programa utilizando solo software libre con la ayuda de una librería para el manejo de nube de puntos.

En esta tesis se propone desarrollar un sistema que se utilizando el dispositivo Kinect como medio de captura y una plataforma de desarrollo como OpenNI [5], se logre capturar información 3D de una escena. Dicha información será luego filtrada, para considerar solo un objeto de interés, y procesada a fin de obtener una geometría de buena resolución y calidad. A su vez, para generar información visual del objeto como la textura, se utilizará la cámara RGB. Este sistema contará con algunas de las estrategias antes nombradas, y proveerá al usuario de una interfaz gráfica que le permita capturar, visualizar y procesar la información de manera sencilla.

Para validar los resultados, se propondrán un conjunto de pruebas o *benchmarks* clásicos en trabajos similares, y se medirá la similitud con objetos 3D diseñados manualmente.

3 Objetivos

El objetivo de este trabajo de tesis consiste en estudiar e implementar algoritmos que, a partir de múltiples imágenes de profundidad obtenidas desde el dispositivo y poder reconstruir objetos reales en tres dimensiones.

Para poder lograr visualizar un objeto tridimensional, primero es necesario filtrar la imagen obtenida o imagen cruda y someterla a una serie de procesamientos. Estos filtros se encargan de disminuir el ruido a través de técnicas de suavizado, súper-resolución, estimación de valores



faltantes entre otros. El resultado es una nube de puntos que ya permite identificar al objeto. A continuación, es necesario aislar el objeto y eliminar el resto de la escena. Para ello se utilizan distintas técnicas de segmentación simples basadas filtros por distancias y color, que permiten discriminar el fondo de la escena como la superficie sobre la cual se encuentra apoyado el objeto. El resultado son una serie de imágenes filtradas de un objeto, las cuales luego se deben componer desde distintos ángulos. Para esto se aplican análisis de la nube de puntos a alinear. Por último, para poder obtener objeto con súper completo se necesita utilizar algún algoritmo de suavizado de superficies 3D.

Para validar los resultados, se propondrán un conjunto de pruebas o *benchmarks* clásicos en trabajos similares, y se medirá la similitud con objetos 3D diseñados manualmente. El proceso descrito anteriormente de presentarán en una aplicación, que vaya guiando al usuario en la parametrización de cada etapa.

4 Cronograma de actividades

Se propone el siguiente cronograma de actividades, algunas de las cuales ya han sido cumplimentadas al momento de presentación de este documento.

Meses 1 y 2: Búsqueda de información bibliográfica.

Meses 2 y 3: Investigación del ambiente definido.

Meses 4, 5, 6, 7, 8: Diseño y desarrollo de los distintos módulos.

Meses 7, 8: Implementación de una aplicación de ejemplo.

Meses 9, 10 y 11: Medición y test sobre la aplicación realizada.

Meses 11 y 12: Informe final sobre el desarrollo.

Aclaración: la documentación del proyecto se realizará a partir del tercer mes y continuará durante todo el proyecto.

5 Bibliografía

[1] Microsoft Kinect. URL: <http://www.kinectforwindows.org>

[2] M. Hornacek, C. Rhemann, M. Gelautz, C. Rother : Depth Super Resolution by Rigid Body Self-Similarity in 3D, IEEE Computer Vision and Pattern Recognition, 2013.

[3] A. Dakkak, A. Husain: Recovering Missing Depth Information from Microsoft's Kinect, Technical report, 2011.



[4] K. Khoshelham, S. Elberink: Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications, Sensors ISSN 1424-8220, 2012.

[5] OpenNI. URL: <http://www.openni.org>

[6] Kinect Fusion. URL: <http://msdn.microsoft.com/en-us/library/dn188670.aspx>

[7] KScan3d. URL: <http://www.kscan3d.com>

[8] Artec 3d. URL: <http://www.artec3d.com>

[9] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, A. Fitzgibbon: KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera, ACM Symposium on User Interface Software and Technology, 2011.

[10] R. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. Davison, P. Kohli, J. Shotton, S. Hodges, A. Fitzgibbon: KinectFusion: Real-Time Dense Surface Mapping and Tracking, IEEE ISMAR, 2011.

[11] Point Cloud Library. URL: <http://pointclouds.org/>

[12] R. Rusu, N. Blodow, M. Beetz: Fast Point Feature Histograms (FPFH) for 3D Registration, Robotics and Automation, 2009.

[13] M. Kazhdan, H. Hoppe: Screened Poisson Surface Reconstruction, ACM Trans. Graphics, 32(3), 2013.