Project Title: ***3D Scanner Using C++***

Md. Kamrul Hasan
Erasmus Scholar on Medical Imaging and Applications
E-mail: kamruleeekuet@gmail.com or m.k.hasan@eee.kuet.ac.bd

Supervised by: Prof. Dr. Yohan Fougerolle, Dr. Cansen Jiang, David Strubel
University Burgundy (uB), Le Creusot, France
E-mail: Yohan.Fougerolle@u-bourgogne.fr

3D scanning has opened a world of opportunities which peruse the real-world objects and acquire its geometrical data and is used to reconstruct 3D model of the object in digital format. These scanning devices have made remarkable impact in various fields such as medicine, health care, research, education, visual arts and many more. In this project, we implement 3D scanner in C++ using Kinect for scanning. Reconstructing and exporting also used for 3D model of a person/ objectives into a 3D printer. Since this project depends on a sensor, it is important to have a data acquisition module that produces desired output. In this project, the Microsoft Kinect v2.0 scanner was used for data capture (both RGB and depth information in 2D).

Objectives of the Projects:

1. Understanding of Kinect, C++ and OpenCV.
2. Understanding of salient features and reconstruction algorithm in OpenCV using C++.
3. To design and implementation software which has ability to perform 3D reconstruction.

Methodology

The overall project works are described in several subsections and blocks that are described below step by step-

3D scanning is the visualization of real-time object in a 3D digitalized format. To achieve the desired project goal, some methods have been used in this project to perform this various operation i.e. computing depth image, detection of key points, extraction of descriptors, feature detection and matching, and finally computing and visualization of the point clouds. Overview of the operations performed in the project shown in Fig. 1. The workflow used to the acquire desired outcome are stated in few steps mentions below-

- ✓ Scanning of the target object using Kinect v2.0.
- ✓ Detection of the key points using SURF detector.
- ✓ Feature Detection and Matching using FLANN based Matcher.
- ✓ Computation and visualization of point clouds.

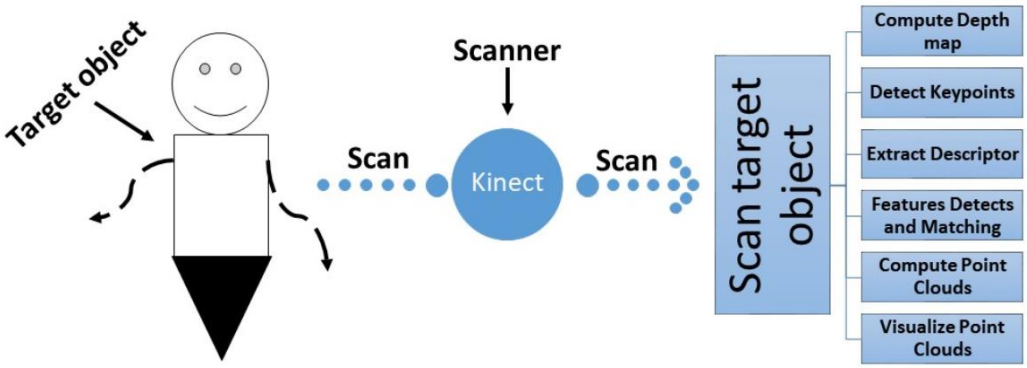


Fig. 1: Overview of the operations performed in the project

Microsoft Kinect v2.0 as shown in Fig. 2. Was used for the scanning purpose which has an RGB sensors (1920 x 1080p), Depth sensors, IR emitters and multi-array microphone with software that provides full-body 3D motion capture, facial recognition and voice recognition capability. Both cameras stream at 30 frames per second. The sensing range of the depth sensor is adjustable between 0.5m and 4.5m.

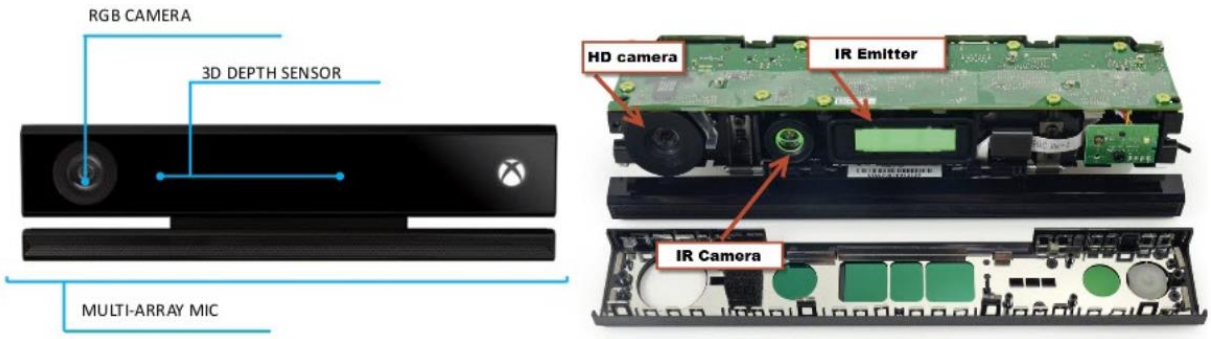
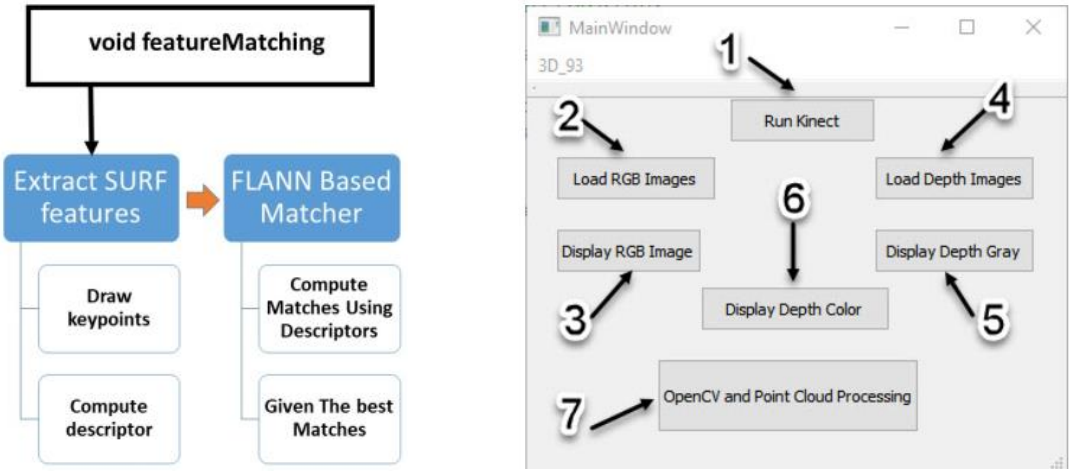


Fig. 2: Pictorial view of Microsoft Kinect v2.0

Speeded up Robust Features (SURF) is a most suitable local feature detector and descriptor in OpenCV C++ (Link) was used in this project as well as Fast Library for Approximate Nearest Neighbors (FLANN) also used for performing a quick and efficient matching by using the Clustering (Link). The block diagram used for the feature detections and matching is shown in Fig. 3 (a).

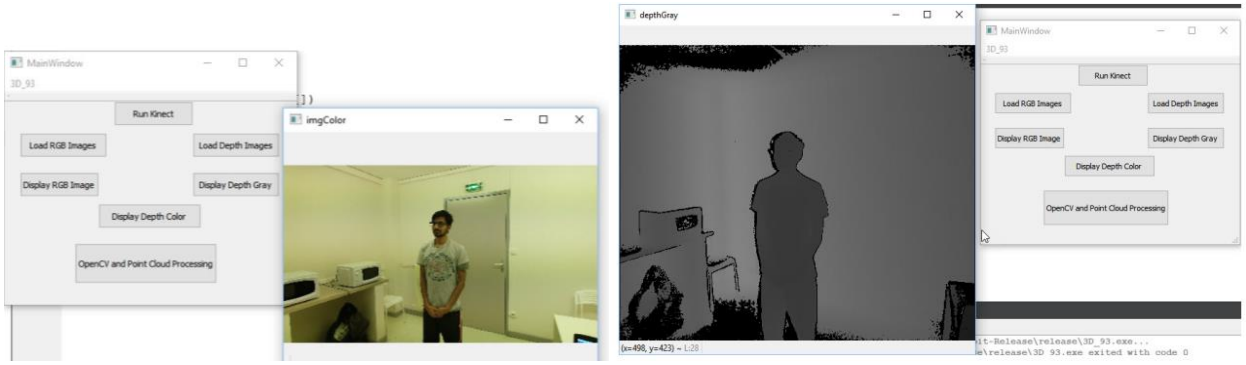


(a) (b)

Fig. 3: a) Feature Detection and Matching procedure b) Shown the different Functionality of the Main Window (GUI/UI)

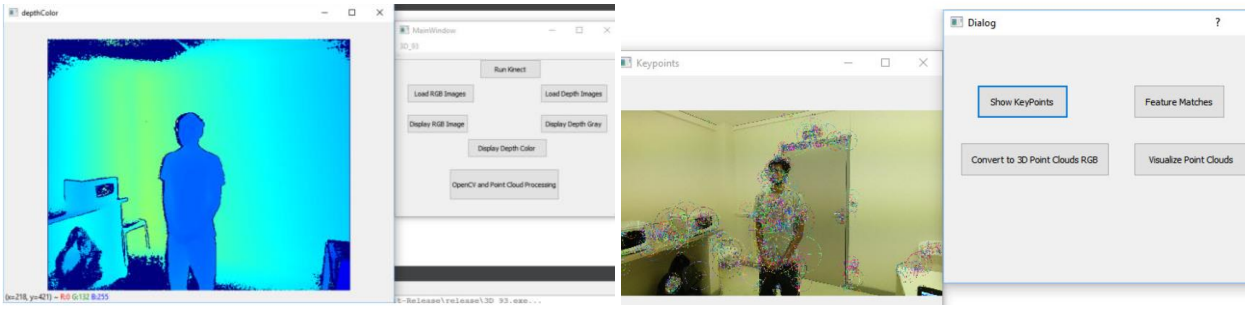
Results and Discussion

Firstly, the Kinect sensors provides the RGB image as shown in Fig. 4 (a) and depth Image as shown in Fig. 4 (b). Combining the both color RGB and depth image, we get depth color image as like in Fig. 5 (a).



(a) (b)

Fig. 4: Preview of a) RGB image b) Depth Image from Kinect sensor



(a) (b)

Fig. 5: Preview of a) Depth color Image b) key points from SURF for matching using FLANN

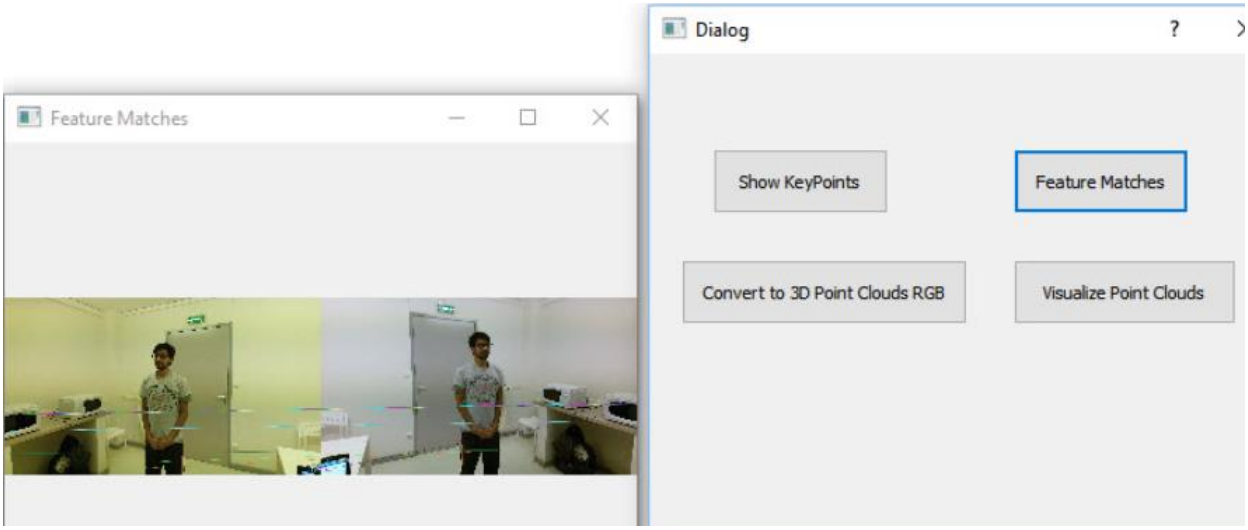


Fig. 6: Preview of the Feature matching using FLANN

After performing the SURF algorithm in OpenCV C++, the best key points are shown in Fig. 5 (b). Then these key points were used for the features matching in FLANN algorithm in OpenCV C++ as like in Fig. 6. And finally, Fig. 7 was our 3D structure of the scanning 2D object after applying some other algorithm like image registrations.

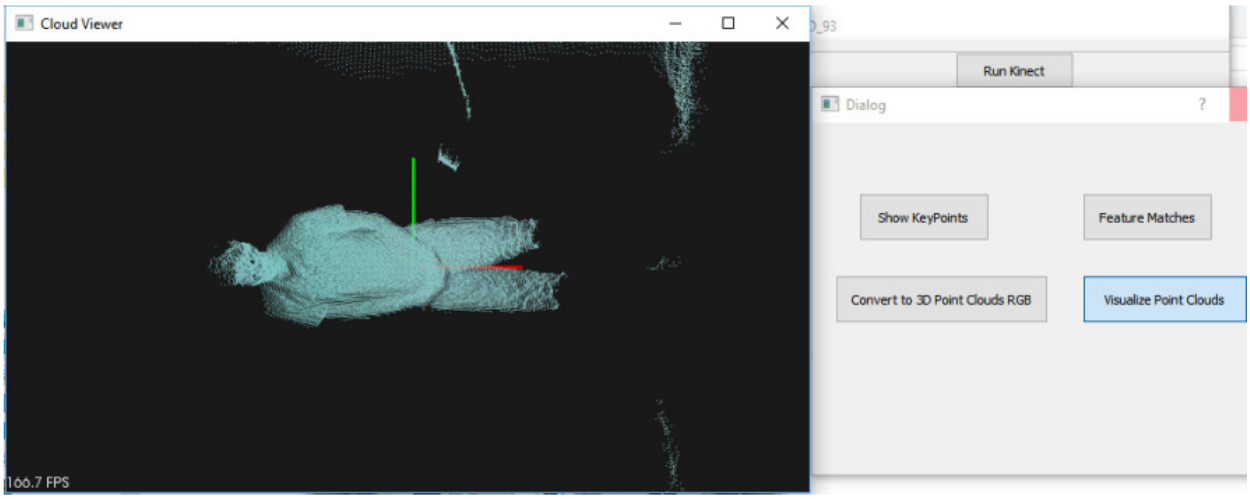


Fig. 7: Visualization of the final 3D structures

Project Partner: Lavsen Dahal, Fakrul Islam Tushar and Ama Katseena Yawson