CQB simulation system,

based on 3D scanning and VSLAM technology

중앙대학교

창의ICT공과대학

소프트웨어학부

Team 01

20164927 김남열

20185659 김혜성

20185228 유희빈

Contents

Table of contents

1. Abstract

Our system is inspired by the Korean Army's simulated combat training and a game called Battleground. It embodies a survival game that can be played at home. To follow the most modern and realistic methodology, 3D scanning and VSLAM technology were used, and the most important issues dealt with were pipelining centered on real-time and socket communication. Rather than listing the intentions and advantages of planning, the paper tries to organize the technical core and detailed implementation, and explain the accumulated experiences in carrying out this project.

1.1. CQB

The letters CQB stand for Close Quarters Battle. In general, It's meaning for a battle which entry into the building or room, or short-range and make combat in order to subdue the enemy. Therefore, CQB simulation system is developed due to training that combat or playing recreational games.

1.2. Indoor Localizing

In order to simulate such a CQB operation, in simple terms, it can be said that there are two elements: a gun and a person. We need to know where this gun and man are. The reason for this is that, in general, the game can be played as long as if it is possible to know a person was hit or not when the gun was fired. In that respect, indoor positioning system technology has a significant weight on the project.

Indoor Localizing is unlike outdoor localizing, it is difficult to use satellite signals such as GPS. (Because it is surrounded by obstacles such as walls.) Therefore, we need to determine the internal position with physical quantities that can be observed anywhere on the earth, such as acceleration, gyroscope, and geomagnetism or use computer vision technology to determine the internal position.

In this project, internal localization using physical quantities is firstly developed, which is relatively easy to implement, and then localization using computer vision technology is developed as an additional subproject if possible.

1.3. Indoor map (3D mesh) construction system using camera

Another key to CQB is in cover. Using these covering wisely is one of the key elements that can lead to victory in battle. Implementing these covers is a key factor in creating a simulation, and in order to implement this, we believe that the construction of a three-dimensional map of indoor space is absolutely necessary. If the location of each user was identified and the trajectory of the bullet was calculated using ballistics at the time of triggering, it would be a normal simulation only by checking whether there is a wall in the trajectory or not to check whether the user gets shotted or not.

In this project, the first 3D reconstruction using IR is performed. 3D reconstruction based on iso surface will be performed, and the entire room will be reconstructed using 3D stitching technology for the reconstructed disparity maps. After that, when basic project development is completed, additional online 3D reconstruction will be performed. This will add a little more realism to the simulation by collecting information such as the destruction of walls and movement of obstacles during the game through the IR attached to the gun.

2. Project Motivation

2.1. Kinect - indoor motion sensing game player

▲ **[picture 01] Kinect-indoor motion sensing game player**

As **[picture 01]**, Kinect is an indoor camera motion recognition game player system and a camera device that is preferred as a stereo input device in the field of computer vision due to various applications. Looking at this, we thought that the game through camera recognition is good, but the space, where we can play, is limited. We wanted to make a game that uses the entire space inside the house that could exist outside the living room.

2.2. "Memories of Alhambra" (K-drama, AR combat simulation)

▲ **[picture 02] "Memories of Alhambra" (K-drama, AR combat simulation)**

As **[picture 02]**, "Memories of Alhambra" is a 2018 South Korean television series. Primarily set in Spain and in South Korea in latter episodes, the series centers on a company CEO and a hostel owner sho get tangled in a series of mysterious incidents surrounding a new and intricate augmented reality game inspired by the stories of the Alhambra Palace. Watching the drama "Memories of Alhambra", We became interested in a battle game between virtual & real space. We thought it will be a good experience to actually make it.

2.3. ROKA's combat simulation

▲ **[picture 03] ROKA's combat simulation**

The ROK military (ROKA) regularly conducts mock field training. Watching the video recorded about this, we could see what would be needed to develop our simulation. However, as can be seen from the record above training is placed on outdoors or a building with thin walls based on numerous wireless devices (GPS etc.) and the measurement devices.

However, we want to implement that with computer vision technology-oriented because we are in Computer Vision Machine Learning Laboratory (CVML). Also, simultaneously, we moved the project domain to a small room where radio signals were scarce. So, the value of this project can be seen as better.

We also thought that if the necessary technologies in this project could be built only with computer vision, it would be of great help in obtaining and rebuilding 3D information in various fields.

3. Introduction

Let's define it as a CQB environment, which is the environment in which all of these systems must operate. In the indoor space, a few players form two teams to define an environment in which the opposing team is annihilated using a gun. Two pieces of information are needed to make it a gaming environment. First, how is interior space constructed? Second, where does each competitor stand, and in which direction is the gun pointed? Two technologies were used to solve these two problems. It is a 3D scan and VSLAM technology.

In the case of the 3D scan, as one of the key technologies of rendering in the field of graphics, papers using laser sketching using mathematical properties of sound function curvature in A volumetric method for building complex models from range image were referred to as the project progress of this project. A paper, "*BundleFusion: Real-Time Globe Consortium 3D Reconstruction Using On-the-Fly Surface Reintegration*", used RGBD sensor background sketching in its history of development.

In the case of VSLAM, it's a core technology of robotics, and although it's a lot of functions similar to odometry, to summarize the focus of this project, it's a system that looks for the R|t matrix between the origin of the world and the camera. This part is based on the open code of ORB SLAM2.

4. Project Conception

4.1. Indoor positioning system using 9-axis sensor with filtering/integration. algorithm

In 3D, the most basic way to find out where the user is and in which direction the gun is pointing (roll, yaw, pitch) is to use a physical quantity that can be grasped anywhere on Earth. For this, you need to get help from a 9-axis sensor. The values given by the sensor are as follows. Acceleration 3-axis for location data, Gyroscope 2-axis for rotation data, Geomagnetism 3-axis for azimuth data

Based on this value, we apply an algorithm that performs filtering and integration and tries to identify the indoor location where the initial value is unknown.

4.2. Stereo-video reconstruction by using 3D reconstruction

In 3.1. we were able to determine the indoor location of the user that did not know it's initial locations. Now we will use the 3D reconstruction technique to figure out the initial locations. First, the whole room is scanned without people through the camera attached to the gun. After that, all of the stereo pairs are rebuilt. After that, we try to build a 3D disparity map of the whole room by implementing/using 3D stitching technology. Based on this 3D map, the game origin, the initial position of each user, and the height of the user's muzzle are additionally inputted. So that the game can be initialized.

4.3. Additional study/project for project improvement (TODO)

From below, it may or may not be developed according to the project progress and the capabilities of the team members. Note that this is an additional matter.

4.3.1. Mobile-real-time-online (updatable) 3D reconstruction system

When checking several papers, we could not find an online 3D reconstruction system based on a moving object and a moving camera at the same time. This is because it is difficult to build with only computer vision technology. (Because it is difficult to match a part on a certain 3D map with an image that was actually taken.) However, since we have stereo image data that includes 9-axis sensor values, we may solve the problem that is difficult to match above. So We will try to solve the top problem.

5. Related Work

Let us introduce our relevant research for the implementation of our system.

5.1. 3D scanning

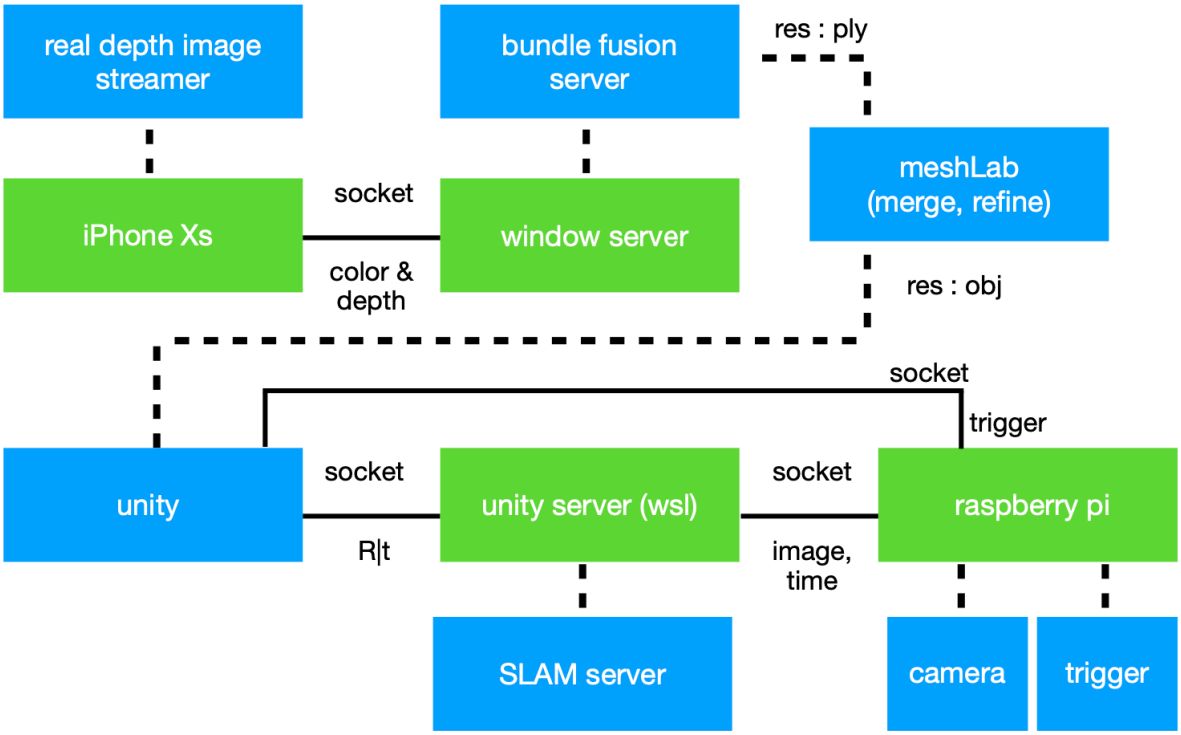
The base of scanning technology is "*A volumetric method for building complex models from range image*". Using laser technology, using continuous and depth data of multiple images, with an object in the center, the 3D mesh is generated around it. The study, which expanded it to scratch it regardless of scale, was "*real-time 3d reconstruction at scale using voxel hashing*", which opened up the possibility of using it in a large space. Then the paper that developed this a little more is "*BundleFusion: Real-Time Globe Consortium 3D Reconstruction On-the-Fly Surface Reintegration*". It is a system that performs the same function, but it is performing better every second. Although the code of this implementation is open to the public, it adds that it is difficult to modify it because it is too vast. The domestic paper "*Modeling Using Sound Function Curvature, Park Mikyung and Lee Euitaek*" are also very noble. The assumptions and models of 3D scanning technology are described in detail in this paper. Further reviewed, but not added to this project, "*convolutional neural networks for 3D shape recognition*", deals with the technology to re-identify what an object is when it is taken in rotation. Here's a glimpse of the difference between the current 3D graphics and the 3D vision: 3D graphics are focused on creating a mesh of thorns by attaching the camera pictures algorithmically, and 3D vision is the same, but using judgment techniques such as neural networks, it doesn't have to be visible.

5.2. VSLAM

For VSLAM, the following paper was referenced. For the first time, I saw "*real-time stereo visual odometry for autonomous ground vehicles*", and this paper deals with tracing cameras attached to mobile vehicles. The key is to identify SURF characteristics and confirm movement through convergence. This feature-based tracking is a key task that is carried out in most odometry, slam papers. However, due to hardware reasons, the two-way camera was not available, so they could not be used directly. The next is the paper "*orb-slam a versatile and accurate monocular slam system*" used for actual implementation. This paper is published up to the third volume, and the machine system used ORB-SLAM2. This is because an attempt was made to use an implementer of indoor slam technology using a neural network called "*Structure-SLAM: Low-Drift Monocular SLAM in Indoor Environments*" based on ORB-SLAM2, but the published software failed to use it due to current-time errors. (The two share the same function signature, so it's compatible.) Additional considerations are "*CNN- SLAM: Real-time dense monocular SLAM with learned depth prediction*". This also uses the neural network for ex-wife re-use, but the implementer was not properly disclosed and could not be used.

6. Project Design

The architecture of our system is divided into two parts.



▲ [picture 04] architecture

In **[picture 04]**, the upper part of scanning space for mesh in the environment and the resulting obj files are entered into two parts: where the user's gun is, where it is headed, whether the bullet hit the opponent, and whether the obstacle was hit.

In the text, we will discuss how these two parts are implemented and the know-how that comes from them. Please check https://github.com/Team-AllyHyeseongKim for everything mentioned.

6.1. Scanning

First of all, for convenience's sake, I tried to carry out the scanning based on the image from Raspberry Pie, but I found that there were many kinds of depth maps, so I placed an iPhone in the pipeline that could obtain an absolute depth map.

The requirement to operate the BundleFusion system as a command system sensor is to input depth images and RGB images, or RGBD. It is important to note that it is an absolute depth image (16-bit, each elt in mm), not a relative depth image. There were many experiments that led to the realization of this fact, which attempted to be executed with a depth map from OpenCV's block matching or a depth map from the implementer of Hierarchical Deep Stereo Matching on High-Resolution Images, but failed to function properly by overlooking the relative value. At the same time, the technique called "rectify" of stereo calibration also results in the loss of many parts of the image if the camera's layout is not '—'. In addition, many images only increase Kelly's error, should be filled with image patterns as much as possible, and unless DSLR is used, motion blur problems can occur because reading Sensors from end to end takes time, so it is recommended to take a picture while stationary. In conclusion, I checked the sensor that has a depth value through an absolute metric and said

it is only possible with a sensor using IR. To that end, the iPhone's real depth sensor was used. (See also store "*iphone-reality-streamer*")

One thing to check is that the iPhone depth image is radial distortion - more visible around - for the color image. This radial distortion can be corrected in object-c, and the processed material was compressed into png and sent using HTTP to the Bundle Fusion.

The server was simply implemented in Node.js to test it before it was processed by the Bundle Fusion. "*real-depth-streamer-server*". In the Bundle Fusion, an image is received asynchronously using a Microsoft-based library called cpprest. - It is not safe for the receiving queue to operate asynchronously, but there was no problem because the image size was too large. - Of course, the code on this side also brought images from HDD for testing and made a version that can be rewritten. "*http-image-server-for-bundle-fusion-scanner*" So the top part of the above architecture, **[picture 04]**, was completed. You can acquire a play file by using this to scan the space. However, due to the nature of the algorithm, it was possible to check the tendency that estimating the location of the camera becomes more and more difficult when the camera is captured. Another option is to use a program called MeshLab to shoot several times and use them. After that, export via obj file, and it can be sourced from unity.

If there is an additional way to replace the iPhone, please use the stereoStreamer-with-recipe. If you're curious about other attempts, please check "*vision-utils-calibrator-depth-map-deblur-odometry-*". Let's see if the other dataset is normal on the BundleFusion. Check "*sens-file-compressor-and-decompressor*" if you want to type in .sens format.

6.2. SLAM

This system works in Ubuntu. It is said that it can be built in Windows, but it is more complicated than I thought, so I chose Ubuntu as the experimental environment. However, to complete the task on Windows pc, we worked on the wsl. The problem is that the network connection is isolated from the host's network because it is connected using a virtual NIC. Please carry out forwarding operation at this time. "*orb_slam2-for-socket-streaming*" uses the setup file of the repository, which can be forwarded by specifying more port numbers in this file, and please check before processing. Also, the ssh display program - xming was used for debugging. After that, we made a socket connection, cueing structure in the same way as shown above. "*rasp-socket-trigger-and-image-sender*" repository is a script that works with raspberry pie, and "*unity- game*" is the final game. In addition, a simple tutorial created a "*unity-python-socket*", and "*socket-receiver-sample-cpp*" repository.

7. Thoughts

7.1. What SW developers should do: understanding the technology of use

This was due to the illusion that the paper's implementations were understandable, numerically applied to any data, and a viable component anywhere, a reliable mass. However, all the codes were dealing with only one implementation for the happy scenario. This is because it is an implement that is designed to demonstrate a paper that is created with the best idea of providing mathematical proofs, so it cannot work in situations other than the assumptions that have entered the proof. In e.g. structure SLAM, we made Manhattan world assumptions and constructed models, and this system would not work if it were full of windings. Of course, as many theorem states, engineering is inevitable because it brings widespread entropy together, but when you make it into software and deliver it to end-users, you need to rethink those assumptions.

In other words, I think we should think about when and why it was made for what, and make sure it meets the coverage of our system and have several alternatives.

7.2. What SW developers should do: understanding the hardware of use

The software should accurately understand the hardware it uses, understand all the number of hardware failures, and respond to any anticipated cases without problems to provide strong and robust integration. However, I should also be aware of the limitations of hardware language. Based on a few papers and showcases, we were counting on IMU to estimate its location. However, it took a little while to find that there was no implementer, and I tried to implement and use it based on the papers. However, a lot of filtering was needed, and nevertheless, it could hardly guarantee normal operation, so it gave up its introduction. Future challenges should be aware of the fact that without something at the library level that has been implemented a couple of times, it is difficult to say that it has been investigated and verified and reliable.

7.3. The importance of the dataset

When it comes to software, it should be considered a very important priority that end users can use it anywhere. However, using the data sets tested, there are quite a few cases where the actual camera input data does not work. I should be aware of these things clearly when I use non-library things. Conversely, it is important to have a dataset, but it is likely to be time to place and rotate the mock object for a while if it is difficult to configure the actual camera and other equipment immediately or if the HW configuration or SW configuration is incorrect. The balance between these two is of considerable importance.

7.4. Thoughts on low-level language

The illusion of c++ is pervasive. "A good programmer should know how to play c++," "c++ is a good language for speed," and so on. I would like to raise a criticism of these two sentences. These sentences are based on assumptions that occur only in very special cases, that there must be clear know-how in implementing a particular technique, a single technology, directly. Otherwise, a low-level language is not suitable under any circumstances. Moreover, it would be 'foolish' to use c++ rather than python when implementing a number of first-time implementations. - No matter how many lyes brushes there are, there are too many risks from build to memory management mistakes. - I think the best judgment is the approach that optimizes only the bottleneck after implementation.

In that sense, I would like to say that the python environment of "*numpy*", "*conda*" is more efficient env. when embodying a paper or new experimental technologies. Of course, it will not be possible to control and connect any low-level hardware, but I don't think there is a problem because that part can easily bring in c++library. Because the basic bases are unified. In the case of c++, it is said that the arrangement, the matrix of Eigen, and the matrix of OpenCV are all different, and although conversion is quite costly, it has better uniformity because most of python is implemented as "*numpy*".

Also, from the standpoint of abstraction. Unlike old cars, modern cars have a lot of electronic equipment (track con trolls, ABS, etc.). These systems require numerous driver's understanding to operate directly. That's why it can't work directly. (Not everyone can be an expert.) I just want to tell people who want to use the computer as a testbed (people who don't want to care about the characteristics of the device) that automated "*cupy*" is more efficient than "*Cuda*".

8. Conclusion

After all, the bottom line for this project is that if you divide up the work that engineers do with research, R&D, and development, three are organized as follows, and one project can do only one of these three. The research should aim to increase performance by adding new assumptions or discoveries based on original research and to prove its happy scenario - simple implementation and experimentation. Based on the original study, R&D should aim to create a reliable software language component for demonstration and integrate to a minimum level to demonstrate the level of the showcase in a controlled environment. The development should aim to integrate already existing, reliable components with perfect and robust integration through good design practices. - I can define architecture as having only reliable components deployed components.

It is regrettable that the project failed to succeed as originally planned, and that some demonstrations were successful. I think the reason is that they tried to carry out all three parts. It is judged that the result was not bad at the R&D level.

9. Reference

1. *"Stochastic Optimization Based 3D Dense Reconstruction from Multiple Views with High Accuracy and Completeness", Chen et al., Journal of Information Science and Engineering 31, 131-146, 2015.*
2. *"A novel 3D dense reconstruction with high accuracy and completeness", Chen et al., IEEE International Conference on Multimedia and Expo Workshops (ICMEW 2013), 2013.*
3. *"Stocastic Optimization Based 3D Reconstruction with High Accuracy and Completeness", Chen et al., The 25th IPPR Conference on Computer Vision, Graphics, and Image Processing.*
4. *"3D Manhattan Room Layout Reconstruction from a Single 360° Image", Zou et al., arXiv:cs.CV/1910.04099, 2019.*
5. *"Real-time 3D reconstruction at Scale using Voxel Hashing", Nießner et al., ACM Trans. Graphics, vol. 32, no. 6, pp. 169:1-169:11, Nov. 2013.*
6. *"A First Look at a Telepresence System with Room-Sized Real-Time 3D Capture and Life-Sized Tracked Display Wall", Maimone et al., in Artificial Reality and Telexistence (ICAT), The 21st International Conference on, nov 2011.*
7. *"Full-Body Awareness from Partial Observations", Rockwell et al., European Conference on Computer Vision (ECCV), 2020.*
8. *"Content-Preserving Image Stitching with Regular Boundary Constraints", Zhang et al., arXiv:cs.CV/1810.11220, 2018.*
9. *"PoseGAN: A Pose-to-Image Translation Framework for Camera Localization", Liu et al., ISPRS Journal of Photogrammetry and Remote Sensing, 2020.*
10. "Wearable Camera-Based Human Absolute Localization in Large Warehouses", *Écorchard et al., Proc. SPIE 11433, Twelfth International Conference on Machine Vision (ICMV), 2019.*
11. *"Beyond Controlled Environments: 3D Camera Re-Localization in Changing Indoor Scenes", Wald et al., European Conference on Computer Vision (ECCV), 2020.*
12. *"Full-Body Awareness from Partial Observations", Rockwell et al., European Conference on Computer Vision (ECCV), 2020.*
13. *"Automatic Panoramic Image Stitching using Invariant Features", Brown et al., International Journal of Computer Vision (IJCV), 2007.*
14. *"Multi-view Convolutional Neural Networks for 3D Shape Recognition", Su et al., International Conference of Computer Vision (ICCV), 2015.*
15. *"Recent Trends of Real-time 3D Reconstruction Technology using RGB-D Cameras", Kim et al., 2016 Electronics and Telecommunications Trends, 2016.*
16. *"Encumbrance-Free Telepresence System with Real-Time 3D Capture and Display using Commodity Depth Cameras", Maimone et al., 10th IEEE International Symposium on Mixed and Augmented Reality, 2011.*
17. *"Real-Time Volumetric 3D Capture of Room-Sized Scenes for Telepresence", Maimone et al., 2012 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2012.*
18. *"Hierarchical Deep Stereo Matching on High-resolution Images", Yang et al., Conference on Computer Vision and Pattern Recognition (CVPR), 2019.*
19. *"Consistent Video Depth Estimation", LUO et al., Signal Interest Group on Computer Graphics and Interactive Techniques (SIGGRAPH), 2020.*
20. *"A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms", Scharstein et al., Proceedings IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV), 2001.*