Viewpoint tracking Display Tutorial

2022 / 8 / 9

YouTube : Medist

GitHub : Medist-1

0.

Before I start, I want to tell you that my code doesn't work as well as it does in my video.

There are some major problems:

1. There are input lags that interrupt your 3D experience.
2. Detecting range is quite limited.
3. Position detecting algorithm is actually incorrect.

So, if you are going to use this codes for products or going to have presentation to others, you may have to choose better algorithm and better hardware.

I really hope that you recognize these problems and won't be disappointed with the result.

1. Requirements

I used Visual Studio 2022, C++, Vulkan, OpenCV, and used laptop camera.

So, I'm not sure if it will work in other environments.

This program tracks tag as a viewpoint. So you should print out the tag that I uploaded.

2. Source Codes

Most of my codes are based on **Vulkan Tutorial** https://vulkan-tutorial.com/.

In this tutorial, you can have multiple pages of tutorials, including how to install APIs.

I recommend you to follow steps up to multisampling chapter.

If you are bored following tutorials, just paste code in that chapter and debug a pile of errors.

If you've done right, you may get the same output as the tutorial,

which means that all APIs are installed correctly.

Now, install one more API, OpenCV, to get camera position and transformation matrix.

Okay. Now all you have to do is to paste my codes and adjust constants to suit your environment.

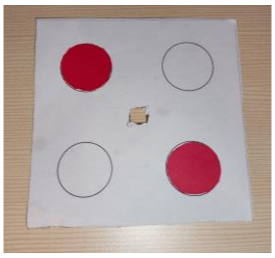
First, paste [main.cpp] and [shader.vert] files. And don't forget to compile shader.

And add [cube.obj] file in models folder, [cube.png] file in textures folder.

Cut out a printed tag and attach it to hard paper.

Color 2 diagonal circles with high saturation color (matte one is better).

And make a hole in the center, so you can see through the camera.

 Like this one.

Build code and see if camera detects 2 colored circles.

You can adjust HSV thresholds through track bars.

The red dot should be at the center of the tag, which will be the point of view.

If it works well, you can fix these thresholds as initial value. (Line 86)

If you can see the cube on screen, and it transforms as the tag moves,

(No matter how it looks), You've done right.

Set window size and camera resolution. (Line 53, 57)

If you want full-screen, you can uncomment [glfwGetPrimaryMonitor()] option instead. (Line 294)

You may have to ctrl+alt+del to get out of full-screen and then reopen it in taskbar.

If you want to end the program, close the cmd window.

I didn’t get how to solve this problem yet ☹

for better performance, comment out camera outputs that are used to adjust HSV (Line 1635)

Attach the tag in front of your camera and see how the object looks.

It may be similar to my work, but slightly different.

(If it doesn’t… that’s too bad. ☹ Check next chapter.)

Now you have to adjust constants to give realism.

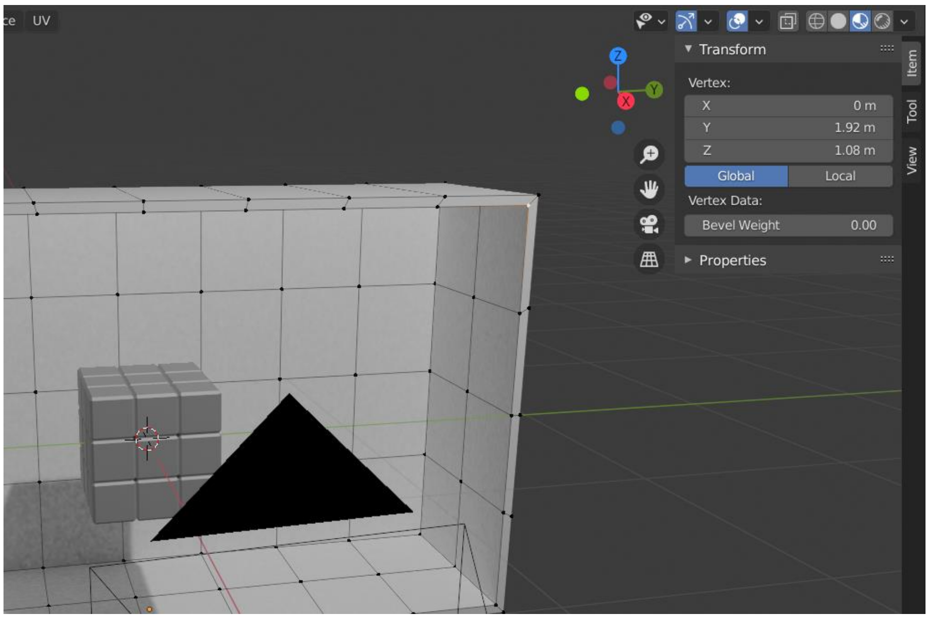
pixel2rad is a constant that converts pixel distance in cam input into radian.

mark_dist is a constant that is related to real distance between 2 marks. (Line 65)

cam_offset_x and cam_offset_y constants are for offset caused by difference between display center and webcam position. (Line 61)

I have my webcam upper center of my screen, so I set cam_offset_y 1.15f

frame_edge_y, frame_edge_z constants are to set virtual frame on model. Set these values proportional to the window size (Line 69)



frame in real model

If you understand this code, you may notice that This method is actually incorrect.

You can use other methods instead to get correct viewpoint position.

You can change model and texture by editing file names. (Line 73)

You can make 3D model and texture by using Blender.

To increase rendering speed, this project didn't use shading option of Vulkan.

Instead, I used the baking option in Blender to give realism to the model.

3. How It works

Calculating center coordinate of two circles and their distance, it calculates camera position, And then simulate projection of sample square. Then by using getPerspectiveTransform() function, comparing original square and projected one, you can get a distortion matrix and send that matrix to vertex shader for coordinates distortion.

4. Problem-solving

When nothing appears on screen / Object looks wield

Fix viewpoint by uncomment (Line 1641) and test again.

If it works well, comment out them again and keep modifying pixel2rad, mark_dist (Line 1628)

Full-screen stops / Can’t exit on full-screen

You may have to ctrl+alt+del to get out of full-screen and then reopen it in taskbar.

If you want to end the program, close the cmd window.

I didn’t get how to solve this problem yet ☹

Model angle is biased

If you see front of the screen, but you get side or upper/lower view of the object, The problem may be caused by difference between camera position and center of the screen. Adjust cam_offset_x and cam_offset_y constants. (Line 61)

Model is stretched / shrank horizontally or vertically

Select window mode instead of full-screen mode. If the model looks normal, set window size (Line 53) same as display size and try again in full-screen mode. If model looks same, set frame_edge_y, frame_edge_z constants(Line 69) proportional to window size (Line 53)

When textures are misplaced

This code modifies coordinates of vertices, which means it doesn’t modify textures on the face.

It can be result in textures misplaced on center of faces. So, I recommend you to subdivide faces.