

Proposal for GSoC 2022

mobile-video-effects

Project Abstract

Add blur and virtual backgrounds to the Jitsi Meet mobile apps.

Project Overview

Description

Currently Jitsi Meet supports virtual backgrounds on the web, but not on mobile, since the implementation needs to be done from scratch. This project would make the mobile apps on par with web.

Prototype

When I worked for Professor Yan Chaokun (doctor of Computer Science) in the virtual reality laboratory, I learned the knowledge related to image recognition and algorithm processing. After I asked him recently and accepted his guidance, I have come up with a scheme idea to realize virtual background blurring and character recognition on mobile terminal. For the following two tasks, I already have highly accomplished code for some of the goals.

Working blur's Implementation

So based on this, we can do the blurred background extraction according to the following steps (this is the idea of Gaussian blur algorithm) :

- A series of contour points matofpoints are obtained by using the rectangular frame of face detection.
- Create a single channel Mat image hole with a gray value of 0, and draw the contour obtained in the previous step on the hole;
- The original image mRgba was processed with Gaussian blur to obtain frameBlur.

- Copy the inner part of the contour in the original mRgba image into the frameBlur using a mask.
- The resulting frameBlur is a blurred image of the face background.

The key is the adaptation of the algorithm to the Jitsi.

Virtual backgrounds' Implementation

The realization process is as follows:

"Opencv get camera" -- "get frames" -- "use mediapipe to get masks In the local" -- "Transform from Opencv to black and white --" Merge matting" -- "Add the custom of virtual background" -- "Merge background and mask -- "finally merge image frames Output to the interface"

Development Process

Community Bonding (May 20 - June 12)

The task at this stage is to get to know the project in depth, and maybe solve some bugs along the way. Do some research on the required technology, talk to community developers and mentors, and change some technical steps or plans.

Coding Phase 1 (June 13 - July 25)

There are 6 weeks in Coding Phase 1 and an evaluation after the phase.

The goal of stage 1 is to complete background blurring on Android or ios. It needs to write some key code: portrait extraction background blur blur algorithm, display output image algorithm, for Android adaptation, ios adaptation

Week 1-3 (June 13 - July 4)

I will write out the fuzzy algorithm and the output display file.

Week 4-6 (July 5 - July 25)

The next step is to use technologies like React Native for Android and ios, and the JITSi will get an exciting release update. We will pass the first assessment at the same time. Once the goals of this phase have been achieved, we review the code and fix any known bugs in the previous code. And writing code should be documented.

Coding Phase 2 (July 25 - September 04)

There are 6 weeks in Coding Phase 2 and an evaluation after the phase. As for its specific execution steps, I have already made a detailed plan and part of the code, so I omit more.

Week 1-2 (July 25 - Aug 8)

We will finish "Opencv get camera" -- "Opencv get camera" -- "get frames" -- "use mediapipe to get masks In the local" -- "Transform from Opencv to black and white --" Merge matting" This may be the main difficulty. They'll apply it to Android and ios, respectively.

Week 3-4 (Aug 8 - Aug 22)

"Transform from Opencv to black and white's picture" Merge matting -- "Add the custom of virtual background" -- "Merge background and mask -- "finally merge image frames Output to the interface" The aim in week 4 is to improve code quality. Maintainability is important to open source projects. Some problems should be solved in this week.

Week 5-6 (Aug 22 - September 04)

Once the goals of this phase have been achieved, we review the code and fix any known bugs in the previous code. And writing code should be documented. And then our work is basically done, and then the final evaluation.

Buffer Time (7 days)

There are 9 buffer days in case something didn't go as planned in the weeks before. I think some of the work might take less than two weeks, so we'll actually move faster and have more fault tolerance

More about Me

Applicant Info

Name: Sen Fang (Wirth)
Email: wirth.fang@foxmail.com
Github: github.com/FangSen9000
Time Zone: UTC+08:00 (China)
Location: Zhengzhou, China
Education: Henan University, Victoria University, Australia, double major in Computer Science and Technology. sophomore.
Telephone: +86 18143465655
CSDN blog: [Wirth's blog](#) (Chinese)

Self-introduction

As for me, I am a pathfinder, I love open source, and really enjoy the atmosphere of GSoC. I have rich experience in image processing under the guidance of my professor Yan in the virtual reality lab. He has helped me a lot in the proposal of this technical solution, and we will continue to keep in close contact. I will add a resume in English in the attachment. It is worth mentioning that I have experience in bytedance youth training camp, which is just like Microsoft Exploration Program or Google Step Program. I am very happy to participate in the JITS! project, I have full confidence in its prospects, and I will continue to contribute to the project even after the GSoC ends.