

# Computer Vision Course Project Milestone Report

## Gambody

Yuyang Rong, Jingyi Huang, Anqi Pang, Jianxiong Cai, Ziyue Li  
*School of Information Science and Technology*  
*ShanghaiTech University*

**Abstract**—In this milestone report we are going to propose the problems we want to solve aiming at our goal of implementing the visual game Gambody. We will clarify the technical approach we are using at present and what we have achieved so far. The remaining milestones including the dates and sub-goals will be given furthermore.

### 1. Introduction

We found a popular game called *Hole in the Wall*. In this game one (or two) player(s) are facing a moving wall with a certain shape of hole, the player have to make certain pose to pass the moving wall or he will be pushed into the water. Such game is interesting but householders can never have the privilege to play in family gatherings or parties since one can hardly find a moving wall nor adequate safety measurements.



Figure 1. Game Logo



Figure 2. Two players in the game posing to pass the wall.

To fix this, we are going to implement a system based on camera so that every one can play this game. In our work, we will be using camera to track real person, it will also give player a bounding box (moving wall). The box and

player's outfit is compared by our system, the return should be a pass (True) or no pass (False).

### 2. Technical Approach

#### 2.1. Getting camera stream

We make use of the Image Acquisition Toolbox to connect camera device to MATLAB and achieve real-time image processing and displaying. In general, we keep capturing an image from the camera stream, doing image processing and displaying the result on the screen as long as the camera is working.

In detail, we first let MATLAB automatically set camera parameters and invoke the camera since the parameters are device dependent. Then a while loop is maintained to implement the primary procedure during the logging time of the camera.

One of the obstacle we are facing now is that we can't add some pause time before the result is showed to the player. But it would be best if we give player some time to wait.

#### 2.2. Crop out body

A simple but useful method has been implemented to detect the body. The principle is that given the camera is fixed, the only area has color changing on the image is the body part.

In detail, we take a photo of the background first, then we subtract it by the images containing body. As the result, we would get a image with only the body part.

Due to the noise caused by camera shaking, we use a gaussian blur and set a threshold to filter out noise. The first version is implmented on the gray value. In other words, the RGB image is first converted to a gray image, then applying the threshold to get the body.

However, that method brings a problem. If the background is purely green, and the body is purely red. In gray images, they are the same. So in version 2, instead of using the gray image, the body is obtained by comparing RGB raw image. If any of the 3 channel changes a lot, it could be part of the body. Future we will use HSV color space which is more robust to the shadows.

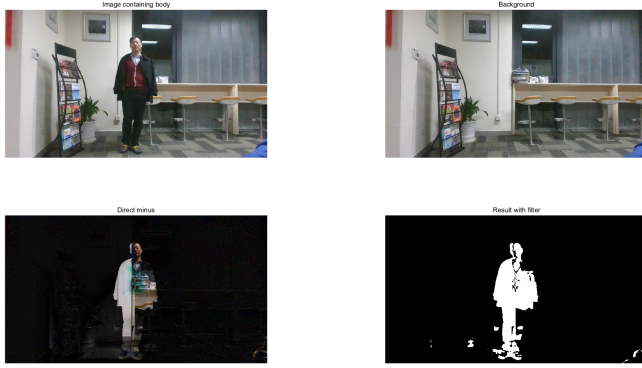


Figure 3. Crop Body

After that, there is still some problems, we get some blobs that are not part of the body. and we might lose some part of the body. So we try to set the biggest (and maybe with the second biggest) connected binary image to be the cropbody.

### 2.3. Generate Masks and Evaluate Results

For we have get the cropped body from previous part, we can generate masks from it. To reduce the workload of evaluate, we use bondingbox to cut the raw image and get a small image which will contain only the body part. And we set different levels for players so the masks should have their own table of difficult levels.

For evaluation part we get two images, one is the players' body and another is the mask. we normalize the two image so it have robustness for different height, then we calculate the correlation of the two pictures and get the biggest corralation value. If the value is bigger than the threshold, then players will pass this level.

## 3. Milestones Achieved So Far

### 4. Remaining Milestones

#### 4.1. Body Pose Recognition

In our first version of Gambody, we use only binary images to determin whether the player pass or not. So a smart player might use the advantage of normalization part and get a pass without a correct pose. Which means, we need body pose recogition from a monocular RGB webcam.

#### 4.2. Background Movement and Multi People

We assume background will not move in the first version of our game but there might be, and maybe there are more than one players. All these will make our assumption fail and so do our code. So we need to deal with multi people and background moving envourment.

### 4.3. Work To Do

To deal with realtime multi-person 2D pose recognition, there are some related works. In our references, there are some methods using CNNs and it is open source. So we decide to use the open source library and rewrite our code by Cpp to speed up.

Also to make our work realtime on the household play stations, we can't use pose recognition all the time. So we will estimate the final image and recognize the pose and skeleton to get scores for player.