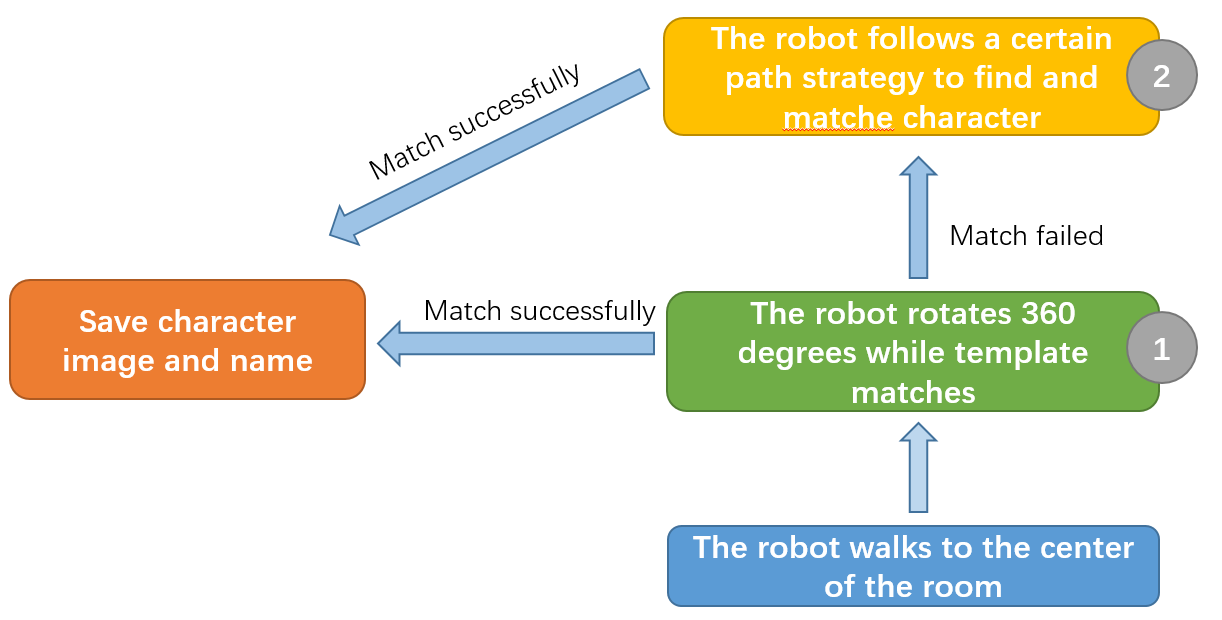
**Character identification：**

The following figure shows the logic executed by the robot in the green room. The design details of the circles 1(Matching strategy) and 2(Path strategy) will be explained in detail later.

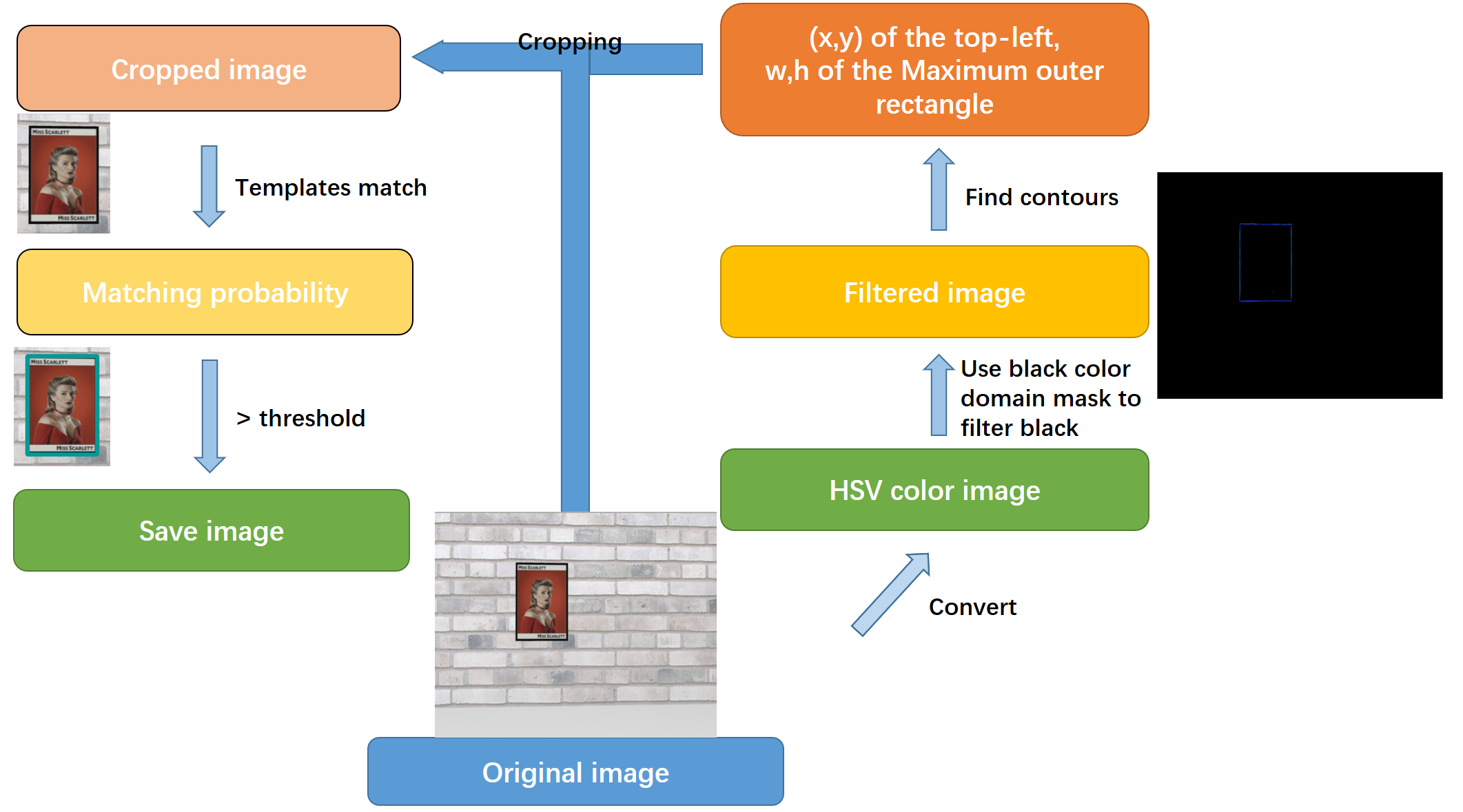
Firstly, the robot enters the room and reaches the centre point of the room. Then, start the template matching strategy and rotate it in place for one round. If the character is found and the matching is successful, the picture and character name will be saved, and if not found, the path strategy will be turned on.



**Matching strategy:**

Based on the observation of the four-character images, it is found that there is a black border around these images, So I designed a matching strategy based on this black border.

The following figure is the logic flow chart of the matching strategy.

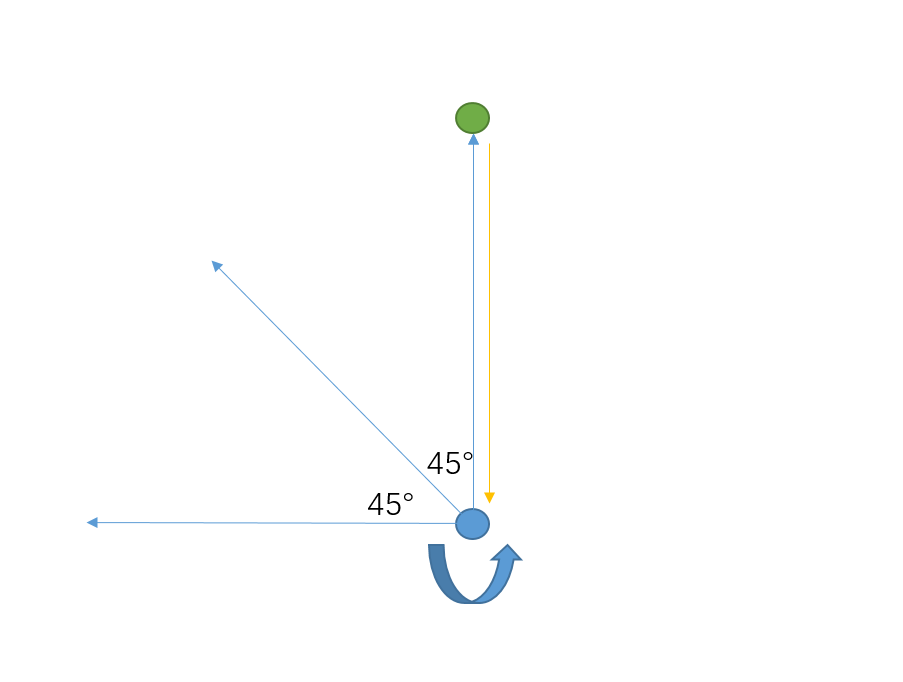


1. First of all, the photo was taken by the robot camera is the original image, which will be converted to the HSV colour gamut in the next step.
2. After that, a black colour gamut mask filters out the black part of the image.
3. Then, we use the ‘cv2.findcoutours’ function to find the contours on the filtered image, and use ‘cv2. boundingRect’ to obtain the coordinates of the top-left point of each black contour and the width and height of the rectangle surrounding the contour. Since the width and height of the character image are proportional, only a similar ratio width and height of black contours can be saved and passed to the next step.
4. In the next step, to remove the interference items, the original image is cropped, according to the top-left point coordinates, w and h.
5. The cropped image is then matched with four templates by ‘cv2. matchTemplate’. If the maximum of the four matching results is greater than the threshold, the character is considered to be found. At the same time, save the image and output the character name of the template corresponding to the maximum matching rate.

**Path strategy:**

We designed two different path strategies to find character images. During the execution of these strategies, template matching is always detected. When the matching is successful, the strategy will stop.

**Strategy 1:**



1. The robot first walks forward a predetermined distance in one direction and rotates 360 degrees after reaching the endpoint.
2. After the rotation is over, the robot will return to the centre point of the room. Then increase the direction by 45 ° and repeat the above operation.
3. After the eighth execution or matching successfully, the strategy will stop.
4. Meanwhile, during walking, if it encounters an object, the robot will retreat, and then return to the centre point to operate in the next direction.

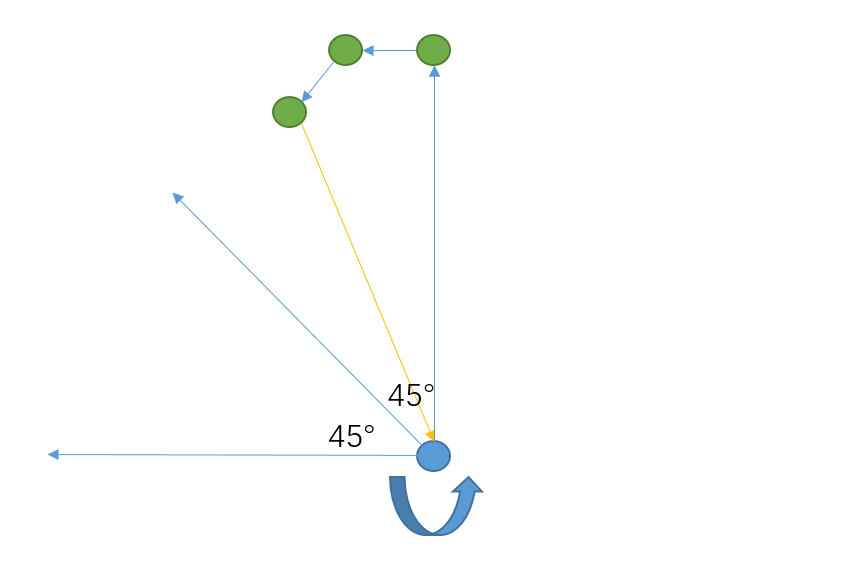
Advantage:

1. Less time used.

Disadvantage:

1. Can't walk to the back of the obstacle.
2. The visible range is limited, the character may not be found.

**Strategy 2:**



1. The robot first walks forward a predetermined distance in one direction and rotates 360 degrees after reaching the endpoint.
2. The robot rotates 90 ° to the left, then walks forward a specified distance and stops to rotate 360 °.
3. Afterwards, The robot rotates 45 ° to the left, stops after walking a distance, and then rotates 360 °.
4. The robot returns to the center of the room, rotates 45 ° to the left, and repeats a, b, c, and d until eight executions.
5. In operation step a, when encountering an obstacle, the robot will take a few steps back, and then execute b. Meanwhile, when encountering obstacles in b and c, the robot will take a few steps and execute d.

Advantage:

1. Have a larger visible area.
2. Can bypass obstacles and see the wall behind them.
3. Higher probability of finding the target.

Disadvantage:

1. Time-consuming.
2. There is a chance that it will get stuck in the dead end.