Door Detection for the Visually Impaired

Derek Sewell (mail@derek.tech)

Nicola Bellotto (nbellotto@lincoln.ac.uk)

1 | Introduction

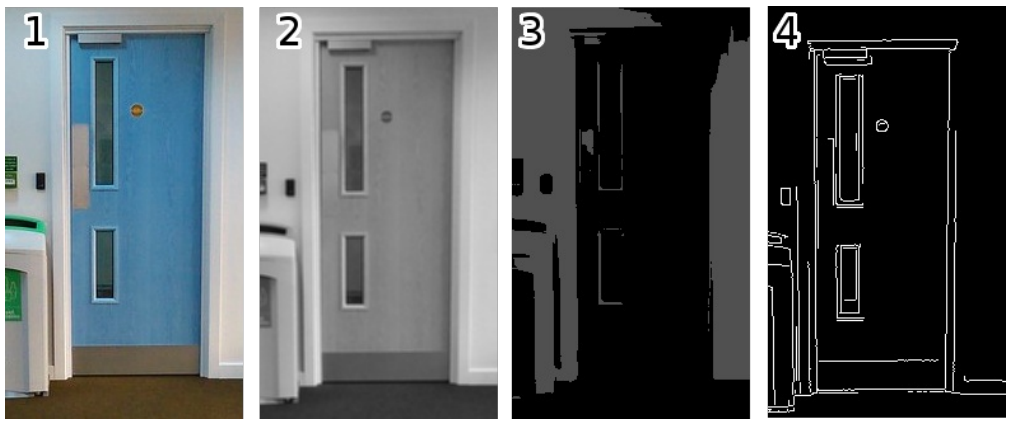
- Detect doors in order to assist the visually impaired with indoor navigation.
- Works with Android smart phones and tablets.
- Uses a novel geometric based approach.

3 | Preparation

The raw image is first broken down into a binary edge map before the door detection algorithm is applied. The process is explained below.

1. Raw image is first converted to grey scale.
2. The image is then blurred to remove noise which would otherwise interfere with the detection process.
3. A threshold is then applied to the image to reduce the number of shades of grey present and thus simplify the detection process.
4. The Canny edge detection algorithm is then applied.

After the above operations the resultant binary image is now ready for the detection part of the algorithm to operate on.



5 | Further Reading

Project source code
github.com/thederek/door-detection

Lincoln Center for Autonomous Research
lcas.lincoln.ac.uk

2 | Advantages

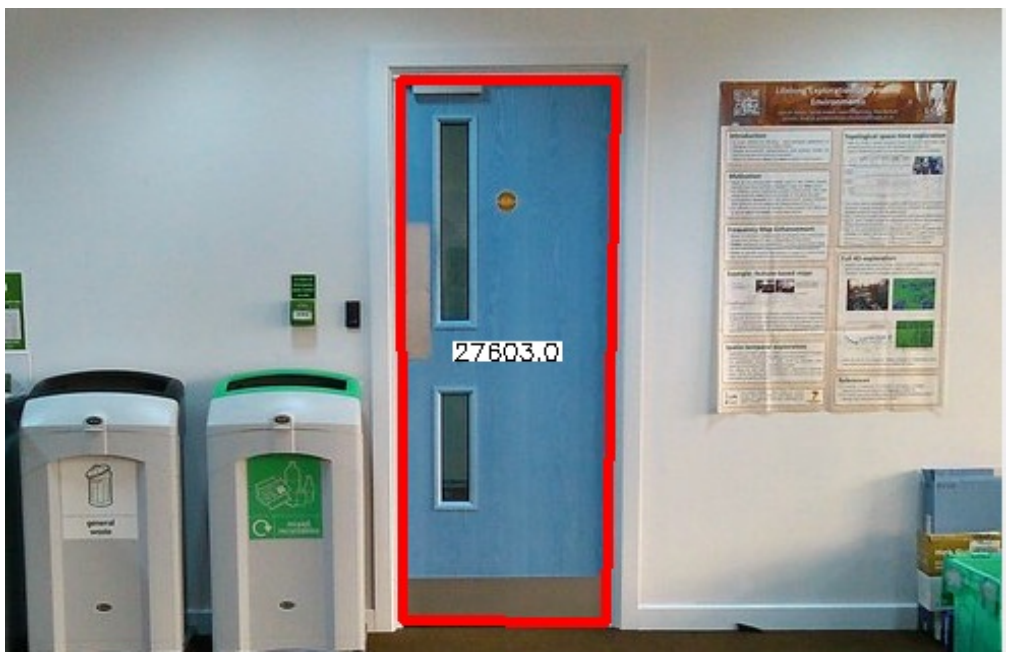
- Increases the accessibility of indoor locations for the visually impaired.
- No additional infrastructure needs to be purchased by the building owner.
- Application is fast enough to be used on a modern smartphone which most people would already have.

4 | Detection

Once the image has been prepared as described in the preparation panel, the detection of the door (if it exists) is commenced.

1. First the contours of the edge map are extracted using the OpenCV function *findContours*.
2. Contours which fulfil the conditions to be a polygon are then added to a separate list. The remaining contours are then discarded.
3. The list of polygons are then filtered based on their area. Any polygons with an area below a certain value are discarded due to them being too small to be a door.
4. The remaining polygons are then filtered based on their number of sides. Any polygons without 4 sides are discarded, thus leaving a list of quadrilaterals.
5. The quadrilaterals are then filtered based on their height to width ratio. Any quadrilaterals that have a ratio outside of the bounds of a single door is discarded.
6. The largest quadrilateral (if present) is then selected as the detected door.

The door is then tracked using the OpenCV function *matchTemplate* until a new door is detected.



6 | Further Work

- Compare the HSV chart of the detected door with the surroundings and other detected doors. Check to see if the values match that of what is normally expected of a door i.e a uniform colour gradient.
- Guide the user to the door using sound cues and tactile sensors that are connected to the users Android device.

Funded by UROS

edeu.lincoln.ac.uk/uros