# Stairways Detection and Distance Estimation Approach Based on Three Connected Point and Triangular Similarity

Md. Khaliluzzaman, Kaushik Deb*
Dept. of Computer Science & Engineering
Chittagong University of Engineering & Technology
Chittagong-4349, Bangladesh
khalil_021@yahoo.co.in, *debkaushik99@cuet.ac.bd

Kang-Hyun Jo
Dept. of EE
University of Ulsan
Ulsan 680-749, Korea
acejo@ulsan.ac.kr

*Abstract* — **Detecting stair region and estimating the distance from a camera to stair in a stair image is the fundamental step in the implementation of autonomous stair climbing navigation, as well as alarm systems for vision impaired people. In this paper, a framework is proposed for detecting the stair region from stair image utilizing some natural properties of stair. One unique property of them is, every stair step's beginning and ending horizontal edge point intersects with two vertical edge points creating three connected point. These vertical edges are stair step's height and its width edge. Another unique property is steps of a stair appear gradually increasing order from top to bottom of the stair in a parallel arrangement. For that initially, directional Gabor filter and Canny edge detector are employed on the stair image to eliminate the influence of illumination and for detecting stair edges. Non-candidate stair edges are removed by performing filtering operation. Then longest horizontal edges are extracted by using a proposed edge linking method on the edge image. After that, a search method is applied for finding stair step height and width edge point at the beginning and ending point of the longest horizontal edges. This operation is performed for detecting three connected points to validate the stair edges. In the next step, increasing longest horizontal edge segments are extracted by comparing x coordinate values of two consecutive edges end points to justify them as stair edges. Finally, from this set of horizontal edges, vanishing point is calculated to verify stair edges from other similar patterns and confirm the detection of stair candidate region. In addition, the triangular similarity is used for distance estimation from camera to stair. The proposed framework is tested using various stair images under a variety of conditions and results are presented to demonstrate the efficiency and effectiveness.**

***Keywords* — *Autonomous climbing navigation; alarm systems; Gabor filter; illumination; three connected point; triangular similarity***

## I. INTRODUCTION

The trend of efficient, appropriate, and advanced intelligent autonomous systems (IASs) has been popular in recent years. Accurate detection and recognition of a particular object in an image is one of the most challenging tasks in the field of computer vision. Detection and distance estimation of the stair region in a stair image is such a task. The stair region detection and estimation of distance from a camera to stair is widely used for autonomous system and vision impaired people. Detection of stair from stair images is difficult because of unusual shape; change of viewpoints, and non-uniform indoor and outdoor illumination conditions while capturing images. Furthermore, to satisfy the criteria of real-time system and AISs, the stair detection method should operate as fast as possible and detect each interest object from the stair image.

This section demonstrates a summary of the various methods that have been tested and implemented for stairway detection and estimation of distance from a camera to stair in the stair image. All the available research works focused on the property that an edge image of a stair contains concurrent long horizontal edges. As the detection and distance estimation of stair region is concerned, scientist and researchers have coined extensive methods to improve the stair detection and distance estimation accuracy. For example, in [1], a framework is developed based on RGB-D images to detect and recognize stairs, pedestrian crosswalks, and traffic signals. To extract the concurrent parallel lines from an image, the Hough transforms is applied [7]. Then the Depth channel is used to recognize pedestrian crosswalks and stairs. Finally, distance between the camera and stairs is estimated for blind users. To detect the stair region from stair image a natural property of stair is used in [2]. The idea of this framework is that stair steps appear in sorted order from top to bottom of the stair by their length. For that, Sobel's edge detector was applied for extracting horizontal edges, and then longest increasing subsequence algorithm employed for extracting longest horizontal edges in a parallel arrangement. Finally, detection of stair region is confirmed by calculating the vanishing point [5][11].

In [3], an algorithm is presented for detecting stair region both in indoor and outdoor environments. The algorithm starts with smoothing the image with Gaussian low pass filter. After that, canny operator and Hough transform are applied to detect the approximately horizontal edges. Finally, use some criteria to detect the stair region. A stair detection system is proposed to help in development of world awareness for vision impaired people through the Smartphone in [6]. A real time stair detection framework is introduced in [8]. This framework uses AdaBoost learning algorithm to detect the stair and estimate the stair step ground plane. A localization method for indoor stairways has been presented in [4]. This method analyzes the stair edges by employing directional filters and planar motion tracking. Then the horizontal edges are extracted by applying Gabor filters [10][12]. From this set of horizontal edge segments, a hypothetical set of edges is extracted by using a correlation method. Finally, a discrimination method is used to detect the ground plane on the basis of behavioral distance measurement. Some constraint such as vanishing point and directional filter were used to detect horizontal edges and stair candidate region in [10][11]. A vision based algorithm is proposed in [12]. This algorithm estimates the parameters using RANSAC algorithm to guide the UGV to climb the stairs automatically under various conditions.

This paper proposes a stair detection framework based on natural and unique properties of a stair. One natural property of stair is, every stair step's beginning and ending horizontal edge (HE) point intersects with two vertical edge's (VE) points. One vertical edge is stair step height and another edge is stair step width. This makes up three connected point (TCP) as shown in Fig. 1(a). This is the key part of the proposed framework. Another natural property of the stair is that stair step's horizontal edges appear gradually increasing order from top to bottom of the stair in a parallel arrangement as shown in Fig. 1(b). Every stair shows these properties when observed the stair front side from a short distance similar to the human vision. In addition, a distance estimation framework is proposed based on Pythagoras and triangular similarity.

The rest of the paper is organized as follows. The proposed stair detection framework is described in Section II. In the next Section, distance is estimated. The experimental result is explained in Section IV. The paper is concluded in Section V.

## II. Proposed Stair Detection Framework

In this section, the proposed framework has been described in details. The proposed framework is divided into six primary steps i.e. (1) applying Gabor filter on stair image and detecting stair edges, (2) filtering non-candidate edges, (3) edge linking and tracking, (4) finding three connected points , (5) extracting increasing horizontal edge segments, and (6) calculating vertical vanishing point and detecting stair candidate region. The proposed framework is as shown in Fig. 2.

### A. *Applying Gabor filter and detecting stair edges*

In order to extract the stair edges properly, noises and shadow effect should be removed from the stair image.

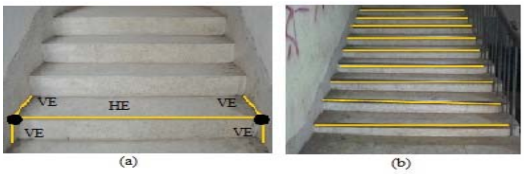



Fig. 1. Some natural property of stair: a) three connected point, and b) longest increasing horizontal edges from top to bottom.

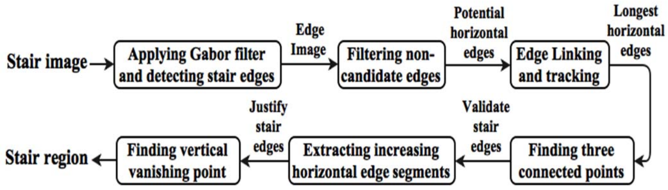



Fig. 2. The proposed stair detection framework.

For that, Gabor filter can be applied using standard convolution method. Initially, the input stair image is converted into a gray scale image for reducing computational cost. Gabor filter is applied to the gray scale image in order to remove the influence of the illumination of stair image and extract the different orientated stair edges. It is a directional wavelet type filter or mask [8]. They consist of a sinusoidal plane wave of a particular frequency and orientation modulated by Gaussian envelope.

Usually, stair is located in a place where natural light falls and lighting condition varies at different daytimes. With appropriate selection of Gaussian variance, orientation, and wavelength, the Gabor filter can remove the influence of illumination variety and shadow effects on an image efficiently and reserve the multi-scale and multi–directional information. The appropriate parameter of a Gabor filer is selected to get the best response from the noise and maintain the stair edges of the stair image. In this regard, various set of values for variance of Gaussian, wavelength, and orientation are chosen to get the most suitable Gabor filter. The general form of Gabor filter is defined in (1), (2), and (3).

$$G_{\lambda,\theta,\varphi,\sigma,\gamma}(x,y) = exp\left(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2}\right)cos\left(2\pi\frac{x'}{\lambda}+\varphi\right) \quad (1)$$

$$x' = xcos\theta + y\,sin\theta \quad (2)$$

$$y' = -xsin\theta + y\,cos\theta \quad (3)$$

where, the coordinate of the pixel is $(x, y)$ and $\lambda$ represents the wavelength of the cosine factor, $\theta$ is the orientation, $\varphi$ is the phase offset, $\sigma$ is the standard deviation of the Gaussian envelope, and $\gamma$ is the spatial aspect ratio.

The standard deviation $\sigma$ is not independent, it depends on bandwidth ($b$) and wavelength ($\lambda$). The relation between $b$ and $\lambda$ is as shown in (4).

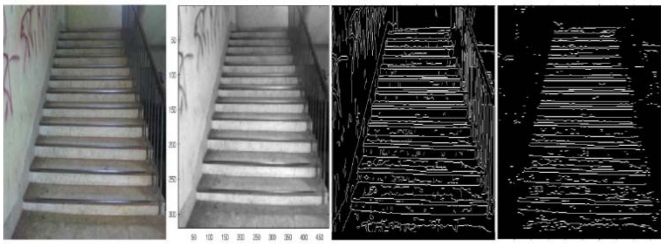
$$\sigma = \frac{1}{\pi}\sqrt{\frac{\ln 2}{2}}\frac{2^b+1}{2^b-1}\lambda \quad (4)$$

If the bandwidth is 1 the relation of $\sigma$ and $\lambda$ is $\sigma = 0.56\lambda$. In this work, it is assumed that all stair edges are horizontal or nearly horizontal and the band width is equal to 1. Fig. 3(b) shows the result after applying Gabor filter on stair image.

According to the natural properties of a stair, the stair steps are horizontally parallel to each other. Hence, the horizontal edges convey information of stair. So, all horizontal edges from the stair image need to be extracted and the potential stair edges need to be found out. Another natural property of stair is that every stair step's beginning and ending horizontal edge point intersect with TCP. So, the vertical edges from stair image also need to be extracted. The Canny edge detector is applied on the filtered gray scale image to extract the edges. The edge detection result is as shown in Fig. 3(c).

### B. *Filtering non-candidate edges*

In this section, potential edges are extracted from the stair edge image. At this stage, the edge image contains different oriented edges. To extract all horizontal edges from the edge image, at first the vertical edges are eliminated from the edge image. After completing the present stage, the edge image mainly contains the horizontal edges. Fig. 3(d) shows the result of eliminating vertical edges from Canny edge image.



(a) (b) (c) (d)

Fig. 3. Processing example of detecting stair edges: a) stair image, b) result of after applying Gabor filter, c) Canny edge image, and d) horizontal edge image.

Besides these, a lot of small and noise edges exist in the edge image. In order to filter the small edges that are not needed, in this paper, a method has been proposed. The idea of this method is that if an edge line is bigger than the THRESHOLD_LINE then the bigger line will be preserved. The procedure to compute the THRESHOLD_LINE size is, length (i) = length of edge i and THRESHOLD_LINE = max (length (i)) /6.

Small edge eliminating procedure

```
for i =1 to No_of_edge
    If  length (i) > THRESHOLD_LINE then
            Preserve the edge i pixels
    else
            Eliminate the edge i pixels
    end if
end for
```

This method eliminates all small and discontinuous edges and saves time for edge linking. Fig. 4(a) shows the result of eliminating small edges from the edge image. Still in the edge image many edges exist that are not part of the parallel concurrent edge. Those edges are eliminated from the edge image as a non-candidate edge, as shown in Fig. 4(b).

### *C. Edge linking and tracking*

At this stage, the edge image contains candidate stair's long horizontal edges only. The long horizontal edges may have small breaks or gaps at some places for various reasons. Still, these pieces represent one horizontal edge. So, small breaks or gaps are required to be filled up. For that, initially horizontal gap of 3 pixels or less are filled up automatically. If the gaps size is more than 3 pixels, then edge linking procedure will be applied. By this procedure, the breaks or gaps in the horizontal edges are efficiently filled in the edge image. Fig. 5 shows the process of edge linking and tracking.

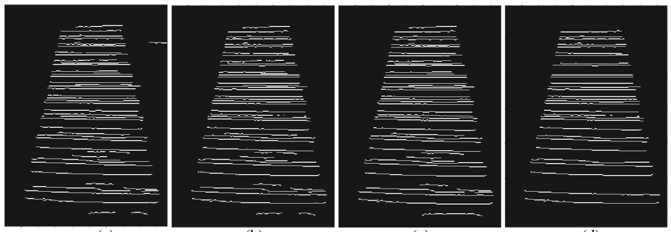
According to the process, two edges are merged if those edges are satisfied the following conditions. For edge $i$ and $j$, if $y_r(i) - y_l(j) < threshold \; and \; x_l(j) > x_r(i)$ then merge edge i$^{th}$ and j$^{th}$ by tracking edge point ( $x_r(i)$ , $y_r(i)$ ) and ($x_l$(j), $y_l(j)$). If $y_l(i) - y_r(j) < threshold \; and \; x_l(i) > x_r(j)$ then merge edge i$^{th}$ and j$^{th}$ by tracking edge point ( $x_r(j)$, $y_r(j)$) and ($x_l$(i), $y_l(i)$). Where, $x_r$, $x_l$, $y_r$ $and$ $y_l$ are $x$ and $y$ coordinate of right and left end points of edge $i$ and $j$. Here, the threshold value is 3.

The result of applying edge linking and tracking procedure in an edge image is shown in Fig. 4(c). Finally, those edges are removed from the edge image whose lengths are less than a threshold is shown in Fig. 4(d). Here max (length(i))/5 is used as this threshold value. This step ensures that the edge image contains the potential long horizontal edges only. Let the number of extracted horizontal edge is N.

### *D. Finding three connected points*

This section focuses on detecting three connected points from edge image. This is done by utilizing a unique property of stair, that is, the beginning and ending points of stair step's horizontal edges intersect with two vertical edge points. These vertical edges are from stair step's height and width, as shown in Fig. 1(a). These three connected points validate the extracted horizontal edges are from stair step edge. For this task, Canny edge image has been used from where longest N horizontal edges are extracted and it also contains vertical edges. According to this property, vertical edges i.e. stair step height and width intersects with horizontal edge's starting and ending points. These intersection points are at the same coordinate of horizontal edge's starting and ending points.

For that, the stair step's height and width vertical edges are searched at the beginning and ending point of the longest horizontal edges for finding the intersection points. If the intersection point is found at the beginning side of the horizontal step edge, then another intersection point is searched with vertical edges at the ending coordinate. In the practical situation, all horizontal and vertical edges in the edge image are not exactly horizontal or vertical. So it is difficult to get the intersection point at the beginning or ending point of the horizontal or vertical edges exactly, as shown in the Fig. 6(a). For that purpose, intersection point is detected with a procedure as shown in Fig. 6(b).



(a) (b) (c) (d)

Fig. 4. Processing example of small edge elimination and edge linking: a) eliminate small edges, b) eliminate non-candidate edges, c) edge linking, and d) potential longest horizontal edges.

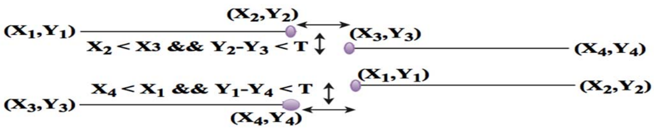
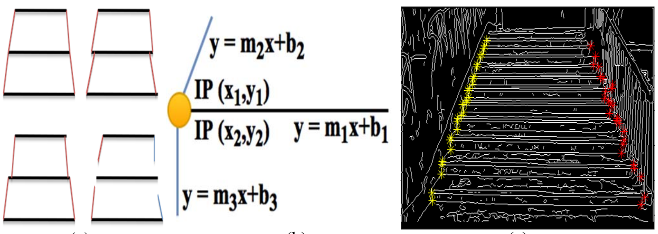



Fig. 5. Edge linking and tracking.




(a) (b) (c)

Fig. 6. Processing example of three connected points: a) different orientation of stair steps height and width edges, b) procedure of calculating three connected points, and c) three connected points in the edge image.

According to the procedure shown in Fig. 6(b), if $x_1 = x_2$ and $y_1 = y_2$ then the intersection point (IP) will be $(x_1, y_1)$. However, all times the $(x_1, y_1)$ and $(x_2, y_2)$ would not be the same because the horizontal and vertical edges are not exactly horizontal or vertical. For that purpose, we will consider $x$ and $y$ coordinate displacement from the starting or ending point of the horizontal edge with a threshold value. The threshold value of $x$ coordinate is $x\_threshold$ =5 and $y$ coordinate is $y\_threshold$=2. According to that, if $x_2 - x_1 < x\_threshold$ and $y_2 - y_1 < y\_threshold$ then the average of $y_1$ and $y_2$ is taken as $y$ and the intersection point will be $(x_1, y)$. Fig. 6(c) shows the three connected points of the vertical and horizontal edges both in beginning and ending side of the horizontal edge.

The approximate numbers of stair steps are measured after completing the detection of three connected points (TCP). In the previous stage, the N horizontal edges were extracted as stair steps candidate edge. If the approximate number of stair steps with three connected points in the stair image is 70% with respect to N horizontal edges, then the extracted N edges are partially considered as stair edge. These longest potential N horizontal edges will be used in the next section for confirming and detecting stair region shown in Fig. 7(a).

### *E. Extracting increasing horizontal edge segments*

This section considers extracting the longest increasing horizontal edge segments of stair from the long horizontal N edges that is found in the previous section and validated by the TCP. In this stage, an important and unique property of stair is utilized that is stair steps gradually appears in increasing order from top to bottom of the stair in the parallel arrangement. If any object edges are in an increasing sequential order, then these edges may be considered as a staircase. For that, the horizontal edges are arranged in increasing order by comparing two consecutive edges's ending point from bottom to top by satisfying the following conditions. For two edges i and j, the conditions are $L_x(i) < L_x(j)$ *and* $R_x(i) > R_x(j)$ where, $L_x(i)$, $L_x(j)$, $R_x(i)$ and $R_x(j)$ are $x$ coordinates of left and right endpoints of the edge i and j. The result of increasing horizontal edge segments is as shown in Fig. 7(b).

### *F. Calculating vertical vanishing point*

In this section, the increasing horizontal edge segments are used to calculate the vertical vanishing point (VP) to verify the horizontal edge segments, whether these edges are extracted from stair or any other similar visual patterns such as train line or zebra crossing. The VP of a stair can be stated as an imaginary point in a linear perspective drawing in which a group of parallel lines are converged. In stairways, the VP can be defined as a point where two handrails of the stair intersect. However, every stair does not have the handrails or some stair have one side handrail. Hence, it is better to construct two virtual handrails instead of depending on real handrails. The right and left handrail is the straight line passing through the right and left end points of the lower and upper edge in the horizontal edge segments. Hence, the VP is the intersection point of these two left and right straight lines.

In Fig. 7(c), the left drawing handrail is the straight line passing through two points $(x_1, y_1)$ and $(x_2, y_2)$. The equation of the straight line is $y = m_1x + b_1$ where, $m = (y_2 - y_1)/(x_2 - x_1)$. Similarly, equation $y = m_2x + b_2$ is derived from right handrail passing through points $(x_3, y_3)$ and $(x_4, y_4)$ where, $m_2 = (y_4 - y_3)/(x_4 - x_3)$. Solving these two equations, the intersection point $(x, y)$ is determined which is the VP of the stair. After that, $y$ coordinate of the VP is checked because the VP for the stair will be imaginary $(y < 0)$ and will reside inside the range of $-2IH \leq YVP < 0$, where $IH$ is the height of the image and $YVP$ is the $y$ coordinate value of the VP. If the calculated VP resides inside the range, the extracted parallel edge segments certainly indicate a stair. On the other hand, those edge segments may be part of some other stair looking object, such as zebra crossing and rail line. They have the similar property as like stairs. The zebra crossing and rail line edges are fully horizontal as like stair steps horizontal edges. However, the $y$ coordinate value of the vanishing point both of the zebra crossing and rail line will be real.

Fig. 8 shows the experimental result of rail line i.e. sample 1 and zebra crossing i.e. sample 2 images. In sample 1, an extracted horizontal edge segment satisfies the TCP criterion for rail line image which is the property of stair edges. However, the calculated vanishing point is (245.05, 85.03) as shown in Fig. 8(e). The $y$ coordinate of the VP is 85.03. The positive $y$ coordinate value distinguishes this image from a stair image. Sample 2 shows that zebra crossing image do not satisfy the TCP criterion. Rail line and zebra crossing VP reside inside of the image as they don't have any slope with respect to the ground plane. So, the $y$ coordinate of the VP of the rail line and the zebra crossing is positive. However, stair has the slope with respect to the ground plane, for that reason stair's VP resides outside of the stair image. Fig. 9(a) shows the example of calculating VP where the $y$ coordinate of the VP is -163.30. Fig. 9(b) shows the stair candidate region in original stair image. The detected stair candidate region is as shown in Fig. 9(c).

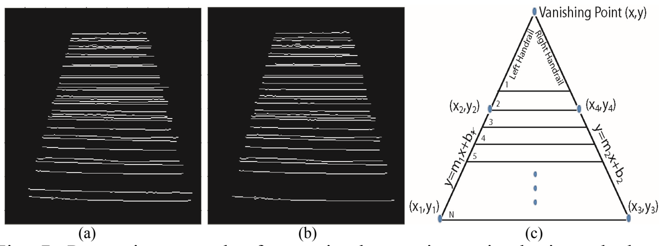



Fig. 7. Processing example of extracting longest increasing horizontal edges: a) longest horizontal edge segments, b) longest increasing horizontal edge segments, and c) calculating vertical vanishing point.

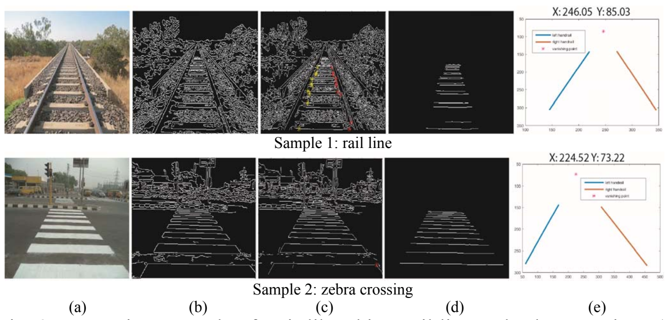



Fig. 8. Processing example of stair like object rail line and zebra crossing: a) input image, b) extracted edges, c) three connected points, d) longest increasing horizontal edges, and e) vertical vanishing point.

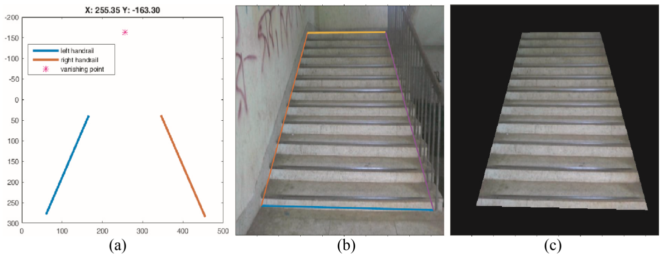


Fig. 9. Processing example of detecting stair region: a) vertical vanishing point, b) detected ROI, and c) stair candidate region.

## III. Estimating Distance from Camera to Stair

Estimating distance from a camera to stair's first step is essential for blind and visually impaired people. By estimating distance they should maintain their walking speed and foot height. It also improves the climbing process of autonomous system in an unknown environment. To estimate the distance from a camera to stair first step, two cameras have been considered in this work. The cameras are placed at the fixed distance with the same height from the ground. The procedure is as shown in Fig. 10. The original distance is estimated from the camera placed at $O$ position. It acts as the first camera. The camera placed at $O'$ is used for calibration. It acts as the second camera in this framework. At first, the beginning and ending point of the stair's first step edge is estimated from the camera position at $O'$. The estimated points are $A'(x_1, y_1)$ and $B'(x_2, y_2)$.The estimated distance between the two points is $A'B' = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$. If the center point of the two estimated points is C then the distance from the left point to the center point of the edge is $A'C = A'B'/2$. The first camera is placed at $O$ from where the distance of the stair first step will be estimated. From camera at $O$ the estimated position of the beginning and the ending point of the stair's first step edge are $A$ and $B$. The distance between $A$ and $B$ is $AB$ and the center point of $AB$ is also at $C$. Let us assume that, the distance between the two cameras is $OO' = d$, the distance from the center point $C$ to the camera at $O$ and $O'$ is $OC = D$ and $O'C = D - d$ respectively.

From Fig. 10, it is seen that $\Delta A'CO'$ and $\Delta ACO$ are similar triangles. And the relation between the distance from center point of stair's first step edge to camera i.e. $CO'$ or $CO$ and the center point to the left end point of stair's first step edge i.e. $CA'$ or $CA$ is inversely proportional. That is,

$$CO' \propto \frac{1}{CA'} \quad or\ CO \propto \frac{1}{CA} \tag{5}$$

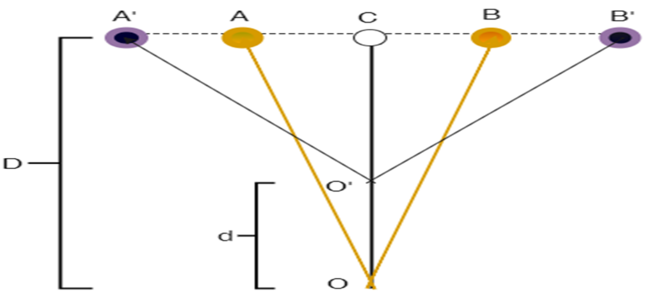


Fig. 10. The proposed framework for estimating distance from the camera.

By using $\Delta A'CO'$ and $\Delta ACO$ similarity, and inverse relation in (5) we can find,

$$\frac{OC}{A'C} = \frac{O'C}{AC} \tag{6}$$

Here, $OC = D, O'C = D - d$

$$\frac{D}{A'C} = \frac{D-d}{AC}$$

$$D = \frac{d}{1 - \frac{AC}{A'C}} \tag{7}$$

$$D = \frac{d}{1-\alpha} \tag{8}$$

where, $\alpha$ is the ratio of $AC$ and $A'C$, $d$ is the distance between the two cameras. $A'C$ and $AC$ is the half of the distance of $A'B'$ and $AB$. The estimated distance between the two end points of the first stair step edge is $A'B'$ and $AB$ that can be found through the image processing. By using similarity property the distance from the camera to the first stair step can be estimated, as described in (8).

### A. Implementation

According to (7), during implementation, the cameras are placed at the fixed distance $d$. The distance $d$ should not be too long or too short. If the $d$ is too long or too short, the linearity relation of $AC$ and $A'C$ could not be maintained. In this work, the distance between two cameras is considered as $d = 0.05\ m$. The cameras should be placed in front of the stair and height of the cameras from the ground should be less than two meters. The angle orientation of the camera should be as closed as the horizontal axis. The framework for distance estimation is as shown in Fig. 11. And the camera setup is as shown in Fig. 12. Due to the shortage of the equipment the images are captured using only one camera moving on a track.

After detecting the stair candidate region, the distance between the camera and the stair's first step is estimated. The processing example is as shown in Fig. 13. The top and bottom stair image at each sample in Fig. 13(a) is captured from the camera placed at $O$ and $O'$ respectively. The beginning, ending and the center point of the stair's first step edge are estimated from the first and the second camera. From these estimated points $AC$ and $A'C$ is calculated. Finally, the distance is estimated by (8).

In Fig. 13, stair sample image 3 is captured from indoor environment and sample image 4 is captured from outdoor environment in different illumination condition for distance estimation. These stair samples are captured from different distances. The distance estimation accuracy from different stair sample images is as shown in Table IV.

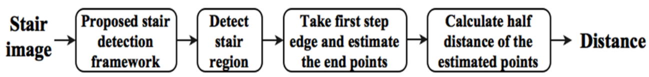


Fig. 11. The proposed framework for distance estimation.

Fig. 12. The camera setup: a) camera with experimental prototype, and b) camera with an experimental prototype at the time of capturing stair image.

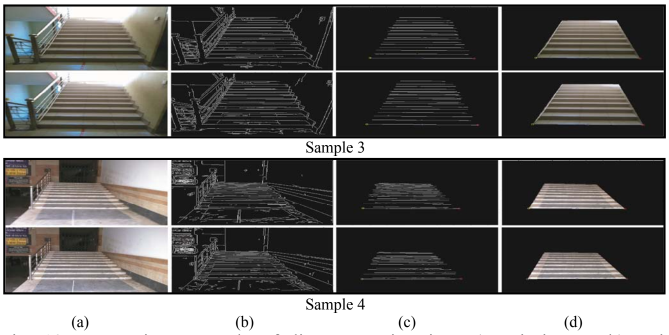



Fig. 13. Processing example of distance estimation: a) stair image, b) stair edge image, c) and d) the beginning, ending, and the center estimating point of first stair step edge in the horizontal edge image and the detect stair region.

## IV. EXPERIMENTAL RESULTS

This section shows the experimental results of various real stair images. All experiments were done on Intel(R) Core(TM) i3-3120M@2.5GHZ processor with 4GB RAM. All the processing was done on 480x320 stair images. The processing was implemented using MATLAB environment. In the experiments, 101 stair images were used to consider various types of outdoor and indoor stair images. These images are captured from the different environment at various illumination conditions such as normal, uneven, and noisy background as shown in Fig. 14. However, some criteria should be maintained during capturing of the stair image. The image should be captured from front side of the stair at a height that should be less than two meters from the ground and not so far away from the stair. The angle orientation of the camera should be as closed as the horizontal axis.

Fig. 15 shows the processing example of stair sample images at different environment and illumination conditions. Input stair image, TCP, longest increasing horizontal edges (HE), vertical VP, and the corresponding output of the processing sample images are shown in Fig. 15(a), Fig. 15(b), Fig. 15(c), Fig. 15(d), and Fig. 15(e) respectively. In Fig. 15, stair sample 5 and sample 6 are straight forward indoor and outdoor stair images captured at day time with normal illumination condition. Stair sample 7 shows the result of outdoor stair with noisy background at day time where some part of stair region is cropped. Stair sample 8 shows the experiment on the indoor stair at day time with noisy background. Stair sample 9 shows the result of the indoor stair at lower illumination condition, and stair sample 10 shows the experiment on the outdoor stair at uneven illumination condition. Each of these stair samples has satisfied the condition of concurrent horizontal edges, TCP, and having a VP with negative y coordinate value. These conditions and run time of stair samples are shown in Table I.

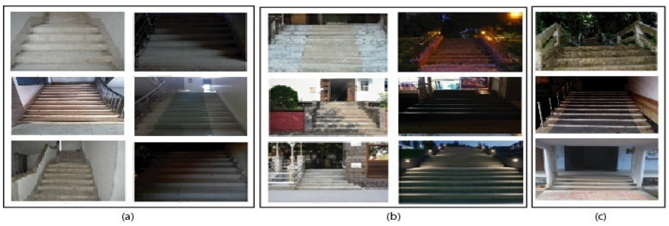



Fig. 14. Various types of stair image: a) indoor stair with uneven illumination, b) outdoor stair with uneven illumination, and c) outdoor stair image with noisy background.

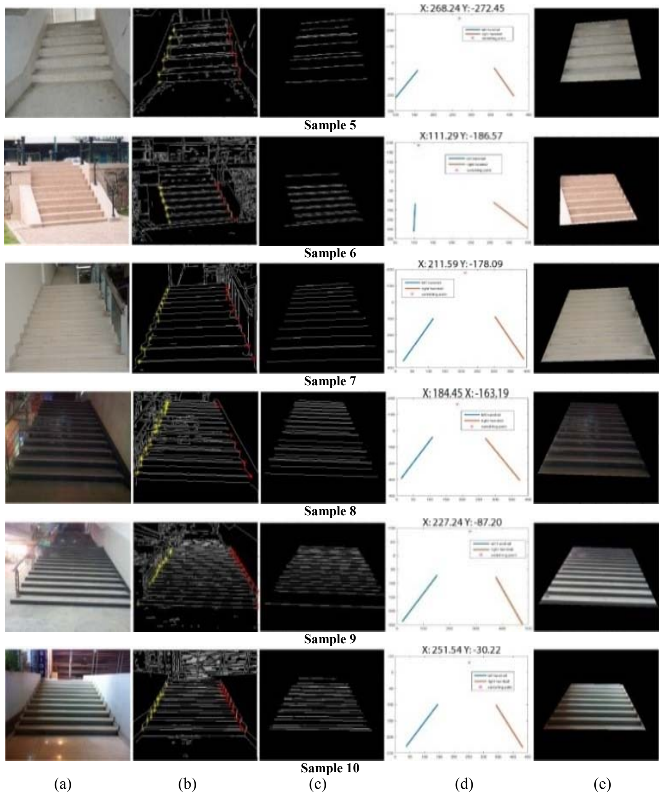



Fig. 15. Processing example for extracting stair region: a) input image, b) three connected points in edge image, c) longest increasing horizontal edges, d) vertical vanishing point, and e) stair candidate region.

TABLE I. THREE CONNECTED POINTS, VERTICAL VANISHING POINT AND RUN TIME OF STAIR SAMPLE IMAGES

| Stair sample | Total HE | Total TCP | TCP (%) | Vertical vanishing point | Run time (s) |
|---|---|---|---|---|---|
| Sample 5 | 11 | 9 | 81.82 | (268.24,-272.45) | 0.058 |
| Sample 6 | 15 | 11 | 73.33 | (111.29,-186.57) | 0.070 |
| Sample 7 | 14 | 12 | 85.71 | (211.59,-178.09) | 0.064 |
| Sample 8 | 28 | 26 | 92.86 | (184.45,-163.19) | 0.071 |
| Sample 9 | 24 | 21 | 87.50 | (277.24,-87.20) | 0.069 |
| Sample 10 | 18 | 15 | 83.33 | (251.54,-30.22) | 0.067 |

In the Fig. 15, stair sample 7 and sample 9 are longest stair compared to other stair samples. In these samples, the beginner stair edges are detected efficiently. After certain distance, stair edges are not extracted properly because of further portion of the sample 7 and sample 9 are narrow, depth and edges are not resolute with respect to the front edges. Fig. 16 shows some processing example at different orientations. Experimental results in Fig. 15 and Fig. 16 show that the proposed framework can detect the stair region from the given stair image efficiently if the framework detects the stair edges effectively.

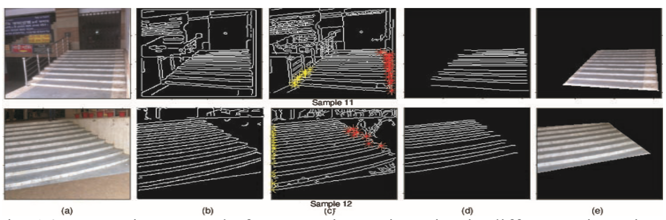



Fig. 16. Processing example for extracting stair region in different orientation: a) input image, b) edge image, c) three connected points in edge image, d) longest increasing horizontal edge segment, and e) stair candidate region.

The detection accuracy is calculated from various types of stair images which are taken from different environment and illumination conditions. The detection accuracy at different environmental conditions is as shown in Table II. The experiment shows higher response to the indoor stairs with normal and uneven illumination condition. It also gives satisfactory response in the outdoor environmental conditions.

The proposed framework has been compared with [2] and [11] with respect to average detection accuracy and computation time shown in Table III.

The distance estimation accuracy from different stair sample images in Fig. 13 is as shown in Table IV. The experimental result of additional distance estimation is shown in Table V. The result reveals that the distance estimation is considerably accurate with an error rate less than 3.15%. Increasing distance from camera to stair decrease the distance estimation accuracy. The reason is that increasing distance from a camera to stair decreases $(1-\alpha)$ which is obvious from the description in (8).

The proposed distance estimation framework has been compared with [4] with respect to distance estimation accuracy and average computational time is shown in Table VI.

TABLE II. DETECTION ACCURACY AT DIFFERENT ENVIRONMENTAL CONDITIONS

| Stair type | Environmental conditions | Total image | Correctly detected | Detection accuracy (%) |
|---|---|---|---|---|
| Indoor | Normal and uneven illumination | 41 | 40 | 97.56 |
| Outdoor | Normal and uneven illumination | 60 | 58 | 96.67 |
| | **Average** | **101** | **98** | **97.12** |

TABLE III. COMPARISON WITH RESPECT TO STAIR DETECTION ACCURACY AND AVERAGE COMPUTATIONAL TIME

| Method | Detection accuracy (%) | Avg. time (s) |
|---|---|---|
| The proposed method | 97.12 | 0.067 |
| [2] | 96.15 | 0.085 |
| [11] | 93.83 | 0.149 |

TABLE IV. DISTANCE ESTIMATION OF SAMPLE IMAGES USING PROPOSED FRAMEWORK

| Stair sample | Actual distance from first camera (m) | Estimated distance(m) | Absolute error (m) | Accuracy (%) | Run time (s) |
|---|---|---|---|---|---|
| Sample 3 | 6.25 | 6.07 | 0.18 | 97.12 | 0.072 |
| Sample 4 | 11.75 | 11.38 | 0.37 | 96.85 | 0.075 |

TABLE V. DISTANCE ESTIMATION USING PROPOSED FRAMEWORK

| SL No | Actual distance from first camera (m) | Estimated distance (m) | Absolute error (m) | Accuracy (%) | Run time (s) |
|---|---|---|---|---|---|
| 1 | 4.5 | 4.46 | 0.04 | 99.11 | 0.073 |
| 2 | 6.0 | 5.87 | 0.13 | 97.83 | 0.073 |
| 3 | 9.25 | 8.98 | 0.23 | 97.08 | 0.075 |

TABLE VI. COMPARISON WITH RESPECT TO DISTANCE ESTIMATION ACCURACY AND AVERAGE COMPUTATIONAL TIME

| Method | Distance estimation accuracy (%) | Avg. time (s) |
|---|---|---|
| The proposed method | 97.61 | 0.074 |
| [4] | 96.65 | 0.078 |

## V. CONCLUSIONS

In this paper, a framework has been presented for extracting stair region from a stair image and estimating distance from camera to stair automatically without any prior information about the position of the stairways in a stair image. These steps are essential to improve the climbing process in an unknown environment for autonomous navigation system. For that, some unique and natural properties of a stair are utilized in this framework. This framework was tested successfully by a group of different stair images with varying styles and different illumination conditions. All the stair images are captured with a height of less than two meters from the ground. The proposed framework detected stair regions of these stair images with acceptable running time and accuracy. The distance estimation from the camera to the stair is also satisfactory with an average accuracy of 97.61%. The experiments demonstrate the effectiveness of the proposed framework. The proposed framework is limited to the detection of traditional stairs with usual shapes. This framework may not give better response for stairs with unusual shapes such as spiral stair. In the future, this work will be extended to detect the stair region from stair images with unusual shapes and implemented for detecting the stair region from downstairs images. And also focus will be given to improve the accuracy of distance estimation.

## *References*


[1] S. Wang, H. Pan, C. Zhang, and Y. Tian, “RGB-D image-based detection of stairs, pedestrian crosswalks and traffic signs,” Journal of Visual Communication and Image Representation, Vol. 25, No.2, pp. 263-272, 2014.

[2] K. Deb, S. T. Islam, K. Z. Sultana, and K. H. Jo, “Stairway Detection Based on Extraction of Longest Increasing Subsequence of Horizontal Edges and Vanishing Point,” Contemporary Challenges and Solutions in Applied Artificial Intelligence, pp. 213-218, Springer, 2013.

[3] S. Shahrabadi, J. M. Rodrigues, and J. H. Du Buf, “Detection of indoor and outdoor stairs,” Pattern Recognition and Image Analysis, pp. 847-854. Springer, 2013.

[4] D. C. Hernández, T. Kim, and K. H. Jo, “Stairway detection based on single camera by motion stereo,” Modern Approaches in Applied Intelligence, pp. 338-347, Springer, 2011.

[5] G. F. McLean, and D. Kotturi, “Vanishing point detection by line clustering,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 17, No.11, pp. 1090–1095, 1995.

[6] S. Carbonara, and C. Guaragnella, “Efficient stairs detection algorithm Assisted navigation for vision impaired people,” IEEE International Symposium on Innovations in Intelligent Systems and Applications (INISTA), pp. 313-318, June 2014.

[7] J. Jos´e, J. Rodrigues, J. and J. du Buf, “Visual navigation for the blind: path and obstacle detection,” International Conference on Pattern Recognition Applications and Methods, Vol. 2, pp. 515–519, February 2012.

[8] Y. Lee, T. Leung, and G. Medioni, “Real-time staircase detection from a wearable stereo system,” 21st International Conference on Pattern Recognition (ICPR), pp. 3770-3773, November 2012.

[9] C. Zhong, Y. Zhuang, and W. Wang, “Stairway detection using Gabor filter and FFPG,” International Conference of Soft Computing and Pattern Recognition (SoCPaR), pp. 578–582, October 2011.

[10] D. C. Hernández and K. H. Jo, “Stairway Segmentation Using Gabor Filter and Vanishing Point”, IEEE International Conference on Mechatronics and Automation, pp. 1027-1032, 2011.

[11] D. C. Hern´andez, and K. H. Jo, “Outdoor Stairway Segmentation Using Vertical Vanishing Point and Directional Filter,” International Forum on Strategic Technology, pp. 82-86, IEEE, October 2010.

[12] Y. Cong, X. Li, L. Liu, and Y. Tang, “A Stairway Detection Algorithm based on Vision for UGV Stair Climbing,” IEEE International Conference on Networking, Sensing and Control( ICNSC) , pp. 1806-1811, IEEE, April 2008.