World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:3, No:9, 2009

# A Valley Detection for Path Planning

In-Geun Lim, Jin-Soo Kim and Chirl-Hwa Lee

***Abstract*—** This paper presents a constrained valley detection algorithm. The intent is to find valleys in the map for the path planning that enables a robot or a vehicle to move safely. The constraint to the valley is a desired width and a desired depth to ensure the space for movement when a vehicle passes through the valley. We propose an algorithm to find valleys satisfying these 2 dimensional constraints. The merit of our algorithm is that the pre-processing and the post-processing are not necessary to eliminate undesired small valleys. The algorithm is validated through simulation using digitized elevation data.

***Keywords*—**valley, width, depth, path planning.



## I. Introduction

THE purpose of a path planning is to enable a vehicle or a robot to move from a start position to a goal position with maximum safety. The algorithms for an optimal path in 2D are mainly based on the graph searching method proposed in [1]. In order to apply graph searching method to path planning, region segmentation is an important preparation process to separate a passable zone and exclusion zone. When we want to plan a path passing through a valley, we need to extract only a passable valley from the area of interest.

To find a valley for path planning, we restrict valleys to those which have a desired depth and a desired width to ensure safe movement. It is a little different from a general concept that valleys are positions where altitude increases rapidly [2]. The depth and width of valley are more important than the variation of altitude for a path of vehicle.

Valleys and ridges are useful geometric features in image processing. Accordingly, many methods have been developed to find ridges and valleys [3], [4], [5], [6]. These methods focused on the points that mark the top of ridges and bottoms of valleys. Unfortunately, the valleys detected by those methods do not guarantee the valley with a desired depth and width. Therefore, it is difficult to apply directly the methods in image processing to constrained valley detection without pre-processing and post-processing.

In this paper, we present an algorithm to detect the constrained valley with a desired width and depth. And then, we can make the valley map using the results. The map represents the valley positions satisfying the constraints. The merit of the algorithm is that the obtained valley guarantees the desired depth and width of it without any pre-processing and post-processing.

In-Geun Lim is with Agency for Defense Development (ADD), Yuseong, Daejeon 305-600, Korea. (phone: +82-42-821-2762; fax: +82-42-821-2224; e-mail: inggy77@naver.com).
Jin-Soo Kim is with ADD, (e-mail: addjskim@yahoo.co.kr).
Chirl-Hwa Lee is with ADD, (e-mail: chhlee@hanafos.com).

## II. Related Works

Ridges and valleys are useful geometric features in image processing. Therefore, many methods have been developed to find ridges and valleys. The key idea of the methods is mainly to use directional derivatives for bending points of terrain [3], [4], [5], [6]. Ridges and valleys in 2D were identified as locations with an extreme height in the direction along which it has the greatest magnitude of its second order directional derivative [4], [5]. Ridges and valleys have a zero magnitude of first order derivative. These algorithms cannot find valleys, whose bottoms are flat.

Besides, ridges and valleys have been also identified in [3], [6] as positive maxima and negative minima of the curvature of the relief's level curves. These algorithms use also the derivatives of the principal curvatures along specific directions.

And methods using slopelines that follow the landscape gradient direction were proposed. Valleys were identified as positions where other slopelines converge to form a channel stream and eventually join in a minimum. But, this method must necessarily separate valleys and ridges.

We would like to take valley features into account for path planning of vehicle. However, valley searching methods in image processing rely on eliminating all the local minima and the planar areas to ensure the continuity of streamlines. Therefore, the pre-processing of smoothing elevation data and the post-processing of eliminating local minima are necessary to apply existing methods to valley search for path planning.

However, we proposed a new method that there is no need to pre-process or post-process and to find valleys with a desired depth and width.

## III. Algorithm

The purpose of our research is to find the valley that guarantees the safe movement. The ultimate goal is to generate valley map by using the results of the proposed algorithm.

### *A. Valley Features for Path Planning*

Valley characteristics for safe movement of vehicle are defined as follows. The valley needs to satisfy two conditions which are desired width, $v_w$ and desired depth, $v_d$ to ensure the space for vehicle movement. The bottom shape is not necessary to search the valleys.

The hatched region $A$ in Fig. 1 satisfies the conditions and the region $B$ does not.

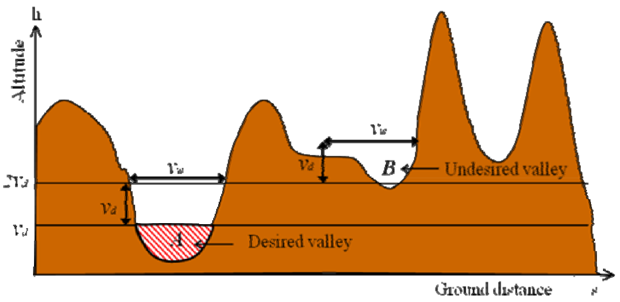


Fig. 1 Valley with desired width $v_w$ and desired depth $v_d$ .

## B. Valley Search Procedure

Let us now consider a valley with width $v_w$ and depth $v_d$ . The valley search is accomplished with three steps. The steps in Fig. 2 outline the procedures for searching the valley which satisfies the conditions.

The input is a $M \times N$ matrix data that represents elevation values at regular intervals. The $T(x, y)$ is a terrain elevation at the position of longitude $x$ and latitude $y$ in the area of interest.

The step 1 is a process to check that a certain position has valley features.

The $V_{l\theta}(x, y)$ is determined by comparing the altitude value of the position $(x, y)$ and those of nearby altitudes with threshold of search elevation level $th_l$ as (1).

$$V_{l\theta}(x,y) = \begin{cases} 0 & \text{if } T(x,y) \geq th_l \\ 1 & \text{else if } T(x_\theta, y_\theta) \geq th_l \text{ and } T(x_{\bar{\theta}}, y_{\bar{\theta}}) \geq th_l \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

This step is processed in every point of input matrix. Therefore, $V$ matrix size is same as input matrix size.

The $th_l$ is the threshold value of the level $l$ , which is defined as (2) where, the notation $l$ is sample level number in altitude domain and the sampling interval is $v_d / 2$ which satisfies sampling theorem.

$$th_l = l \frac{v_d}{2} \quad \text{for } l = 1, 2, \ldots, n \quad (2)$$

$n$ is the number of levels in altitude domain, which is determined by (3).

$$n = \text{int}\left[ \frac{\max_{x,y}\{T(x,y)\}}{\frac{v_d}{2}} \right] \quad (3)$$

$$\begin{cases} x_\theta = x + \frac{v_w}{2}\sin\theta, & x_{\bar{\theta}} = x + \frac{v_w}{2}\sin(\theta + \pi) \\ y_\theta = y + \frac{v_w}{2}\cos\theta, & y_{\bar{\theta}} = y + \frac{v_w}{2}\cos(\theta + \pi) \end{cases} \quad (4)$$

for $\theta = 0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}$ .

Where $(x_\theta, y_\theta)$ and $(x_{\bar{\theta}}, y_{\bar{\theta}})$ are neighboring positions at a distance of $v_w / 2$ in the directions of $\theta$ and $\bar{\theta} = \theta + \pi$ , respectively. Therefore, the positions are calculated by (4). The relationship of the points is as shown in Fig. 3.

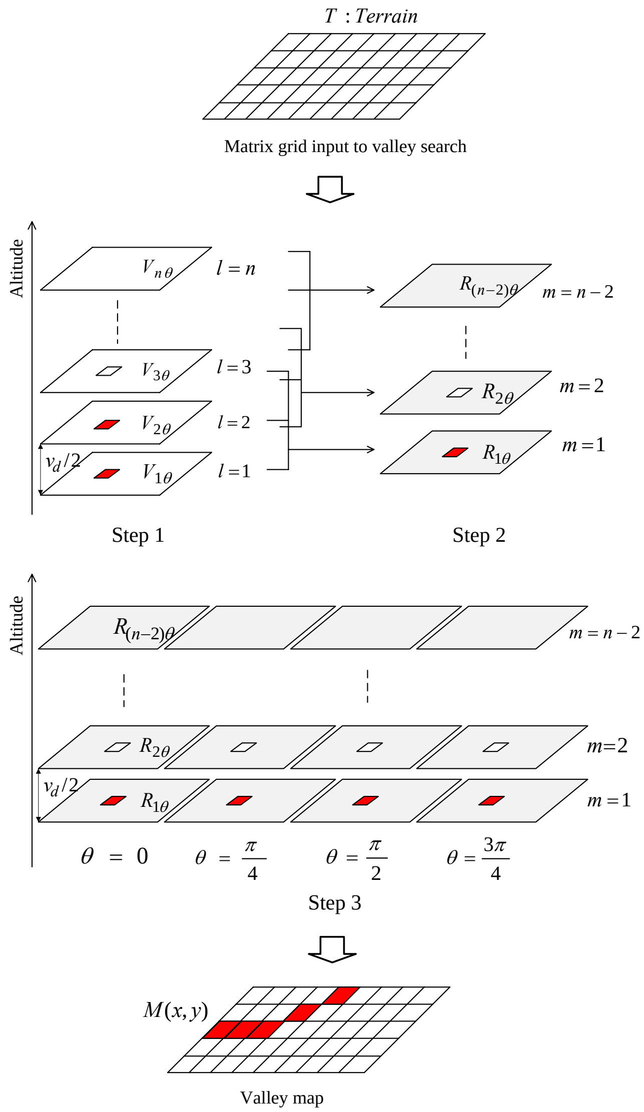


Fig. 2 Valley search procedure.

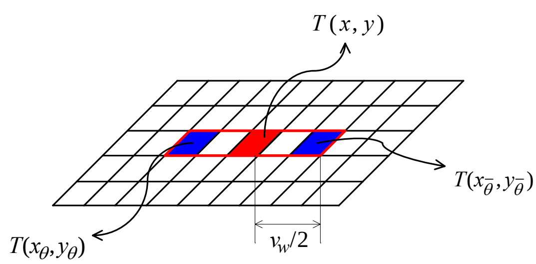


Fig. 3 The relationship of points.

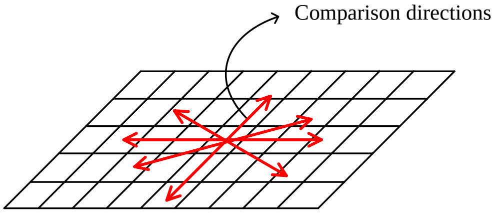


Fig. 4 Comparison directions.

The continuous terrain altitude is sampled at the intervals of $v_d / 2$ from zero to the maximum height in the area of interest, as shown in step 1 of Fig. 2.

To compare $T(x, y)$ with neighboring terrain in 8 directions, we set $\theta = 0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}$ as shown in Fig. 4.

If the number of samples is $n$, $4n$ each of $V$ matrices are generated by step 1.

The step 2 is the process to find locations satisfying a second condition which is a desired depth $v_d$. The $R_{m\theta}(x, y)$ is determined by comparing $V_{(l+1)\theta}(x, y)$, $V_{(l+2)\theta}(x, y)$ of upper levels with $V_{l\theta}(x, y)$ of the current level as (5).

$$R_{m\theta}(x, y) = \begin{cases} 1 \quad if \quad V_{m\theta}(x, y) = 1 \quad and \quad V_{(m+1)\theta}(x, y) = 1 \\ \qquad and \quad V_{(m+2)\theta}(x, y) = 1 \\ 0 \quad otherwise \end{cases} \tag{5}$$

If the number of $V$ levels is $n$, the number of $R$ levels is $4(n-2)$ and $m = 1, 2, \ldots, n-2$. It is illustrated in step 2 of Fig. 2.

The step 3 is the process to determine desired valley. If the position $(x, y)$ is desired valley, at least one of all $R_{m\theta}(x, y)$ equals one as (6).

$$M(x, y) = \bigcup_{m,\theta} \mathrm{R}_{m\theta}(x, y) = \begin{cases} 1 & if \ valley \\ 0 & otherwise \end{cases} \tag{6}$$

$where \quad m = 1, 2, \ldots, n-2, \quad \theta = 0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}.$

Finally, the valley map $M(x, y)$ is acquired as Fig. 2, and it represents regions satisfying the conditions. The map is a binary image that represents whether each cell in a map is the desired valley or not.

## IV. Experiment Results

Using the algorithm above, we present experimental results in this section. The first results appear in Fig. 5. We derive the terrain in Fig. 5(a) from a digital terrain elevation data to form a shading map according to elevation value. Applying the proposed algorithm, we create the valley map. The valley map is overlapped with raster map in Fig. 5(b). Fig. 5(c) represents an altitude profile from point A to point B on Fig. 5(b). The yellow colored regions in Fig. 5(b) indicate the area surrounded by neighboring hills. As you see, the marked region faithfully avoids the ridges of the terrain and indicates a valley. The searched result is not a line but an area with a width as shown in Fig. 5(b).

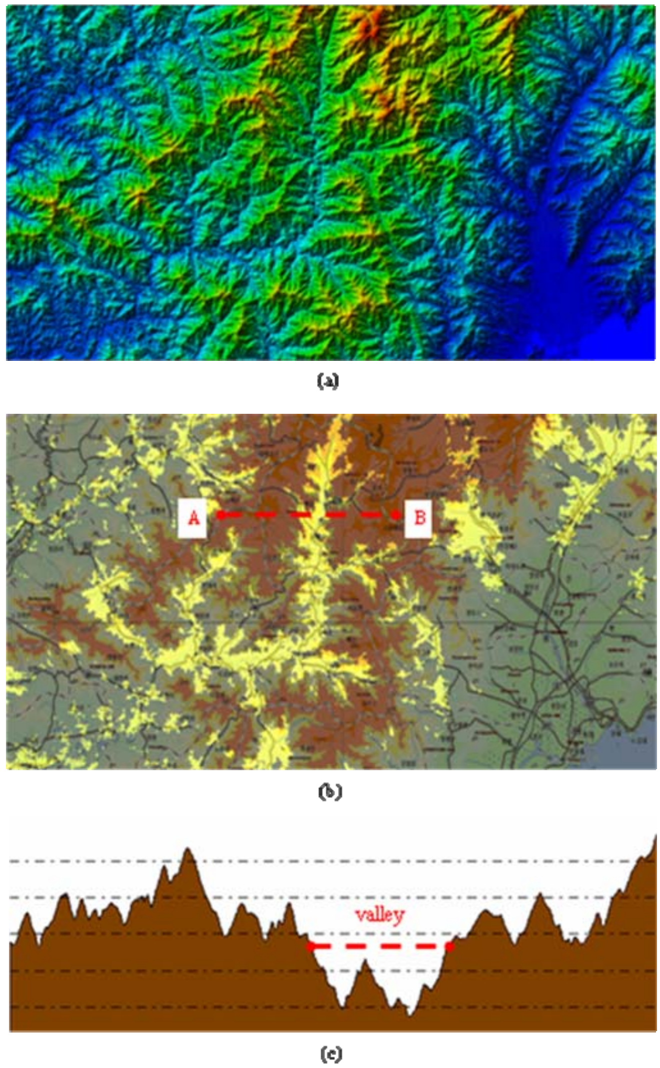


Fig. 5 (a) Digital elevation data (b) simulation result: the marked region indicates valley (c) altitude profile from A to B.

To confirm the effect of width value $v_w$, we made a simulation with a various valley widths. We can see the effects of valley width in Fig. 6 where $v_w = w$ in Fig. 6(b), $v_w = 2w$ in Fig. 6(c) and $v_w = 3w$ in Fig. 6(d). Then, narrower valleys are detected in Fig. 6(b) than those of Fig. 6(c) and Fig. 6(d).

## V. Conclusion

We have proposed an algorithm to detect the constrained valley and generate the valley map.

We showed the usefulness of the proposed algorithm through simulation using digitized terrain data. The simulation results showed that the proposed method can find the valleys very usefully.

The merits of the proposed method are as follows:

1) The pre-processing and the post-processing are not necessary to find the constrained valley.

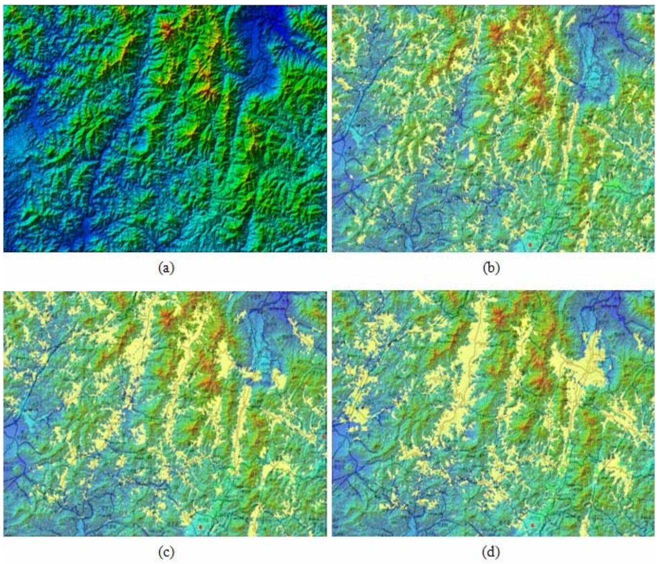


Fig. 6 The Effect of valley width $v_w$ for (a) digital elevation data. Experiment results with (b) $v_w = w$ (c) $v_w = 2w$ (d) $v_w = 3w$ .

2) The result guarantees the valley with a desired depth and width.

The algorithm can be useful in not only path planning for a vehicle or a robot but also the region segmentation.

## REFERENCES

[1] E. W. Dijkstra. "A Note on Two Problems in Connection with Graphs", Numerische Mathematic, 1959.

[2] Antonio M. Lopez, Felipe Lumbreras, Joan Serrat, and Juan J. Villanueva, "Evaluation of Methods for Ridges and Valley Detection", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 327-335, April 1993.

[3] D. Page, A. Koschan, M. Abidi, and J. Overholt, "Ridge-Valley Path Planning for 3D Terrains," in *Proc. IEEE Int. Conf. Robotics and Automation ICRA06,* Orlando, FL, pp. 119-124, May 2006.

[4] R. Haralick, "Ridges and Valleys on Digital Images," *Computer Vision, Graphics, and Image Processing*, vol. 22, no. 10, pp. 28–38, Apr. 1983.

[5] D. Eberly, R. Gardner, B. Morse, S. Pizer, and C. Scharlach, "Ridges for Image Analysis," *J. Math. Imaging and Vision*, vol. 4, no. 4, pp. 353-373, Dec. 1994.

[6] J. Gauch and S. Pizer, "Multiresolution Analysis of Ridges and Valleys in Grey-Scale Images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 6, pp. 635-646, June 1993.