Robust Algorithm for Motion Analysis Based on Active Tubes

Ryo Furukawa, Masakazu Imai, and Takeshi Uno
ryo-f@is.aist-nara.ac.jp

Department of Information Science,
Nara Advanced Institute of Science and Technology,
8916-5 Takayama, Ikoma, Nara, Japan

Abstract

Determining motion of objects is very important and difficult problem in the field of computer vision. Many researches have been studied in finding motion of objects from image sequences. We presented a new model, named Active Tubes, to find the motion of non-rigid objects from an image sequence. Active Tubes analyzes the temporal context in a spatio-temporal solid using an energy minimizing model like Snakes. You can consider Active Tubes as one kind of accumulation of Snakes along the time axis using temporal context.

In this paper, we will point out disadvantage of Greedy Algorithm and propose an extend algorithm to deform Active Tubes. In the former paper, we used the Greedy Algorithm to converge Active Tubes. The Greedy Algorithm determine new node deterministically. The Algorithm is weak to existence of noise. To improve the robustness against to noise, our new algorithm determines new node randomly. Experimental results show that our algorithm is as fast as Greedy Algorithm, and more robust from noise.

1 Introduction

Determining motion of objects is very important and difficult problem in the field of computer vision. Many researches have been studied in finding motion of objects from image sequences.

The simplest method to find the motion of objects is subtraction of continuous images [1][2][3]. It is hard to find correct motion because of the existence of noise or change of lightening. There is another method to find the motion of objects by subtracting the image sequences and the background image. The background image is estimated by accumulating the image sequence, taking a image without moving objects or something other method. Another method is based on optical flow. Optical flow shows which pixel moves. According to the optical flow, we can estimate the motion. Pattern matching methodology is also meaningful to track features. There are a lot of researches in this field.

Occlusion makes the motion of object ambiguous and most methods proposed so far avoid this problem. When the moving object has same intensity as the background image at some points, as often occurs, it is impossible to separate the moving objects from the background. Human beings can easily estimate the motion while occlusion or object having the same intensity as the background. Yamamoto used spatio-temporal images (x-t planes) to find and track the motions of objects [4]. He pointed out that a moving region yields a shape like a tube in the spatio-temporal solid. He also showed that the problem of occlusion can be solved by finding the correspondence between tubes. He recovered occlusion using a method based upon the fluid mechanics.

We presented a new model, named Active Tubes, to find the motion of non-rigid objects from an image sequence [5]. Active Tubes analyzes the temporal context in a spatio-temporal solid using an energy minimizing model like Snakes [6]. You can consider

Active Tubes as one kind of accumulation of Snakes along the time axis using temporal context. We used the spatio-temporal (ST) solid to perform Active Tubes. An input image sequence should be handled as one image solid. In ST solids, a locus of the moving object consists a 3-D region. The region seems like a tube. This makes us free from the problem of feature correspondence. The problem of extracting and tracking the motion of objects in an image sequence can be replaced as a problem of segmenting the ST solid. In [5], we used the Greedy Algorithm [7] to minimize the energy function. The Greedy Algorithm is fast but often falls into the local minima because of existence of noise.

In this paper, we will point out a disadvantage of Greedy Algorithm and propose an extended algorithm to deform Active Tubes. The Greedy Algorithm determine new node deterministically. On the other hand, our algorithm determines new node randomly. Our algorithm is as fast as Greedy Algorithm, and more robust from noise.

2 Active Tubes and Greedy Algorithm

To detect and track motion of objects, we can use spatio-temporal pace (ST space). In ST space, an image sequence is represented as a solid which is called an ST solid, and an object in the image sequence is represented as a region of the ST solid (Figure 1). We can track moving objects by picking out the regions.

Active Tubes is a model to find surfaces of the regions. It is figured like a tube in ST space, and attracted by surfaces of regions. If placed near a surfaces, Active Tubes deforms and converges to the surface. Figure 2 shows how Active Tubes deforms and converges to a surface.

Since we use energy minimizing method to deform Active Tubes, we have to define an energy function. We represent a point on the surface of Active Tubes by $v(s,t)$, where t is a parameter of temporal domain, and s is a parameter of spatial domain having range of

$[0,1]$ along the surface of the Active Tubes at time t. The basic form of energy function is defined as

$$E^*_{tube} = \int_T \int_0^1 \{E_{int}(v(s,t)) + E_{ext}(v(s,t))\}\, ds\, dt \tag{1}$$

where E_{int} is the internal energy, E_{ext} the external energy. The function of the internal energy is keeping Active Tubes smooth and shaped like tubes, and that of the external energy is pulling nodes of Active Tubes toward a surface of region of an ST solid.

From now on, we will represent Active Tubes and the energy function by discrete forms so that we can implement Active Tubes on computers. We represent Active Tubes by a set of parameterized nodes. Figure 3 shows the structure of Active Tubes. A node is pointed by two indexes, s and t. Discrete form of the energy function is

$$E_{tube} = \sum_T \sum_S \{E_{int}(v_{s,t}) + E_{ext}(v_{s,t})\}. \tag{2}$$

The discrete form of internal energy is expressed as

$$\begin{aligned} & E_{int}(v_{s,t}) \\ = \;& \alpha_{s,t} \left(|v_{s,t} - v_{s-1,t}| - \bar{d}\right)^2 \\ & + \beta_{s,t} \left|v_{s-1,t} - 2v_{s-1,t} + v_{s+1,t}\right|^2 \\ & + \gamma_{s,t} \left(|v_{p,t+1} - v_{s,t}|^2 + |v_{p,t-1} - v_{s,t}|^2\right) \\ & + \delta_{s,t} \left|v_{p,t-1} - 2v_{s,t} + v_{p,t+1}\right|^2 \end{aligned} \tag{3}$$

where $v_{p,t-1}$ is the nearest node to $v_{s,t}$ at time $t-1$, $v_{p,t+1}$ is the nearest node to $v_{s,t}$ at time $t+1$, and $\bar{d}$ is the average distance between nodes. $\bar{d}$ is recalculated for each iteration. We can control the shape of Active Tubes by adjusting $\alpha_{s,t}$,$\beta_{s,t}$,$\gamma_{s,t}$,and $\delta_{s,t}$. $\alpha_{s,t}$ controls the the continuities and $\beta_{s,t}$ controls the the elasticity in the spatial domain. $\gamma_{s,t}$ controls the the continuities and $\delta_{s,t}$ controls the the elasticity in the temporal domain.

The external energy is a kind of potential energy which is defined for each point in the ST solid only by its position. To attract nodes of Active Tubes, the energy must have

minimal value at a surface of regions in an ST solid. The following form is often used as external energy:

$$E_{ext}(v_{s,t}) = w_{edge}\left\{-\left|\frac{\partial}{\partial x}I(x,y,t)\right|^2 - \left|\frac{\partial}{\partial y}I(x,y,t)\right|^2\right\} \tag{4}$$

where $I(x,y,t)$ is the intensity of pixel in an ST solid at the location of $v_{s,t}$, and w_{edge} is the weighting coefficient. Using this form, however, there may be no gradient except at surface of regions. Therefore, this form doesn't have much ability to control nodes which is far from surfaces. To cover this problem, moving average of (4) or convolution of (4) and a certain function are often used as external energy. Using this method, you can make energy gradient to control nodes at places far from surfaces, and in addition, you can also smooth the external energy.

There are several methods to converge Snakes. Active Tubes, accumulation of Snake along the time axis, can be obtained using similar method to Snakes. In [5], we used Greedy Algorithm to deform Active Tubes because of the amount of calculation. Greedy Algorithm searches nodes which minimize the energy function sequentially and iteratively. Figure 4 illustrates how this algorithm works. To deform Active Tubes, the following procedure is repeatedly executed for each nodes. The energy function is computed for the current location of $v_{s,t}$ and each of its neighbors assuming that the other nodes doesn't move. Then, node $v_{s,t}$ is moved to the location which gives the minimum of the energy function, and the next node is processed in the same way. This method is fast, having computational complexity $O(mn)$, where n is the number of the nodes and m is the size of neighborhood. But it is easily affected by noise as described in the next section.

3 External Energy and Noise

In images of natural scenes, there are noise and many small objects like dusts, textures,

scratches, and blobs. As described in the following, these objects are very harmful to get the correct result of Active Tubes. We will call all of these objects 'noise'.

The noise tends to have high value of gradient magnitude, which we often use to calculate external energy function. So, (4) often has very small value at the positions of noise. If the function of gradient magnitude is unimodal around noise, you can suppress the effect of the noise to a certain extent by using moving average or convolution of (4). Figure 5 shows this case. However, noise usually makes several peaks of high gradient magnitude values. In this case, using moving average or convolution can't suppress the effect of noise so efficiently. Figure 6 shows this situation. As a result, the external energy at the locations of noise often has very small value. We will call these local minima 'pits'.

Pits strongly attract nodes of Active Tubes, because they have small external energy. If there are a pit near a node, the node is attracted by the pit. Once the node moves into the pit, the node often can't get out of it. The reason for this is as follows. The value of external energy at the bottom of a pit is often the smallest in all of its neighbors. So, if a node exists at the bottom of a pit, the current position often gives the minimum of energy function value in the neighborhood. Then, the current position will always be selected as the next position, because the point having the minimal value in the neighborhood is selected as the next position in Greedy Algorithm.

We will call this situation 'capture'. Captured nodes can't converge to the edge of the target objects. So, this phenomenon is very harmful to get the correct result. Figure 7 shows this situation.

4 Randomized Greedy Algorithm

In this section, we introduce randomized Greedy Algorithm (RGA) to deform Active Tubes. This method is an application of annealing to Greedy Algorithm, and more robust

against noise than Greedy Algorithm.

4.1 Procedure of Randomized Greedy Algorithm

Procedure of RGA is almost the same as Greedy Algorithm. In RGA, the next position of each node is selected by probabilistic method, instead of deterministic method. To do this selection, we define probability function for each candidate of next position of a node.

In the proposed method, like in Greedy Algorithm, the next location of a node is chosen from the points of its neighborhood. We call these points CNPs (candidates of next position). We calculate probability function values for each CNP as follows. First, we calculate the energy function value for each CNP. Then, we determine the priority numbers for all CNPs, depending on the order of energy function value. For example, at the CNP giving the minimum of energy function, the priority number would be zero, and at the CNP where energy function is the second least, the priority number would be one. Finally, we calculate the probability function values using the following form.

$$P(x, y, t) = (1 - a)a^{n(x,y,t)} \tag{5}$$

where a is probability factor, and $n(x, y, t)$ is the priority number of the CNP. Probability factor is a number which controls randomness of deformation of Active Tubes. The larger this factor is, the more randomly the Active Tubes deforms.

The next position of a node is selected according to the probability function (5). The probability that a CNP of the node will be chosen is represented by this function. The sum of probability function of all CNPs is following.

$$(1 - a) + (1 - a)a + (1 - a)a^2 + \ldots + (1 - a)a^{N-1} = 1 - a^N \tag{6}$$

where N is the number of CNPs. So, sometimes, we can't determine the next position, because the total probability (6) is less than 1 by a^N. In this case, the node remains at the current position.

Figure 8 shows how our algorithm works. In this figure, A is the point giving the minimum of the energy function, B the point giving the second minimal value, and C the point giving the third minimum. The probability of A is $1-a$, and that of C is $(1-a)a^2$.

If we set probability factor to 0, probability function at the candidate point giving the minimum of energy function would be 1, and 0 at the other points. So, the point having the minimal value would be always chosen as the next location. In this case, RGA works as same as Greedy Algorithm.

4.2 Advantages of Randomized Greedy Algorithm

RGA is an extended method of Greedy Algorithm, and it inherited the advantages of Greedy Algorithm: speed and simplicity. Greedy Algorithm can be easily modified into GRA, and the computational complexity of RGA is $O(mn)$, which is as good as Greedy Algorithm.

In addition, RGA is more durable against capturing than Greedy Algorithm. The reason for this is as following: A node being captured in a pit of energy function can be illustrated as figure 9. If the pit is the point giving the minimum of energy function in the neighborhood, the node can't move with Greedy Algorithm. But, with RGA, the node may move to every point in neighborhood in next iteration. So, if probability factor is large enough and the pit is shallow enough, the node may be able to escape from the local minimum of energy, and energy minimization may be able to continue. Figure 10 illustrates this situation.

4.3 Convergence of Randomized Greedy Algorithm

In Greedy Algorithm, we can detect convergence of Active Tubes very easily. The

number of nodes that move in each iteration reduces as Active Tubes converges. If the number of these nodes are fewer than a certain threshold number, we can consider that the Active Tubes has already converged. However, in RGA, we can't use this method, because many nodes continue to move even after Active Tubes has globally converged to the target surface.

To solve this problem, we are considering reducing probability factor gradually as Active Tubes converges. This function hasn't been implemented yet. In the current implementation, convergence is decided by human.

5 Implementation and Experiment

To demonstrate the efficiency of RGA, we implemented this algorithm in C on IRIS Indigo workstation. We experimented both Greedy Algorithm and RGA for a sequence of images. The size of the ST solid was $100(x) \times 100(y) \times 30(t)$. It contained a moving basket ball and other objects, and there were much noise in these images. Figure 11 shows two images in the ST solid. Both experiments were done with the same energy weighting coefficient and the same initial positions: $\alpha = 1, \beta = 1, \gamma = 0.2, \delta = 0.8, w_{edge} = 1.4$. Figure 12 shows the initial position. In RGA, the probability factor a was set to 0.3. The size of neighborhood was 7×7. The number of nodes of the Active Tubes at the start of calculation was 1083 in both experiments.

Figure 13 shows the result of Greedy Algorithm. With Greedy Algorithm, several nodes were captured by a pit from noise. The Active Tubes converged after 36th iteration with some nodes captured. Fig.13 (b) shows the final position of the Active Tubes. The execution time was 31 seconds for the first 10 iterations, and 83 seconds for all 36 iterations.

Figure 14 shows the result of RGA. At first, several nodes were captured by a pit as

shown in Fig.14 (a). The nodes was able to escape from the pit and the Active Tubes continued to deform. The decision whether the Active Tubes converged or not was made by human. Fig.14 (b) shows the final result at 42nd iteration. The execution time was 32 seconds for the first 10 iterations, and 99 seconds for all 42 iterations.

As shown by these experiments, RGA gave better result. RGA prevented capturing while Greedy Algorithm couldn't. The performance of RGA was almost as good as Greedy Algorithm.

6 Conclusion

We pointed out a problem of Greedy Algorithm and proposed a new algorithm. The problem is due to the fact that noise makes local minima of the external energy functions. These local minima often attract and capture nodes of Active Tubes and are very harmful to get correct results. We proposed an algorithm similar to annealing method for optimization. The proposed algorithm is more durable against this problem, and is as fast as Greedy Algorithm. We implemented this algorithm and demonstrated the efficiency of this method.

References

[1] R. Jain and H. H. Nagel, "On the analysis of Accumulative Difference Pictures from Image Sequence of Real World Scenes", IEEE Trans. on PAMI, PAMI-1, pp. 206-214, 1979

[2] M. Yachida, M. Asada and S. Tsuji, "Automatic Analysis of Moving Images", IEEE Trans. on PAMI, PAMI-3, pp. 12-20, 1981

[3] S. Tsuji, M. Osada, and M. Yachida, "Tracking and Segmentation of Moving Objects in Dynamic Line Images", IEEE Trans. on PAMI, PAMI-2, pp. 516-522, 1980

[4] Masanobu Yamamoto, "Motion Analysis by Visualized Locus Method", Tras. of Information Processing Society of Japan, Vol. 22, No. 5, pp. 442-449, 1981. (In Japanese)

[5] Minoru Takahata, Masakazu Imai and Saburo Tsuji, "Determining Motion of Non-Rigid Objects by Active Tubes", Proc. of 11th ICPR, Vol.1,pp.647-651,1992.

[6] M. Kass, A.witkin, and D.Terzopoulos, "Snakes : Active Contour Models", International Journal of Computer Vision, Vol.1, No.4, pp.321-331, 1988.

[7] D. J. Williams and M. Shah, "A Fast Algorithm for Active Contours", Proc. of Third ICCV,pp.592-595,1990.

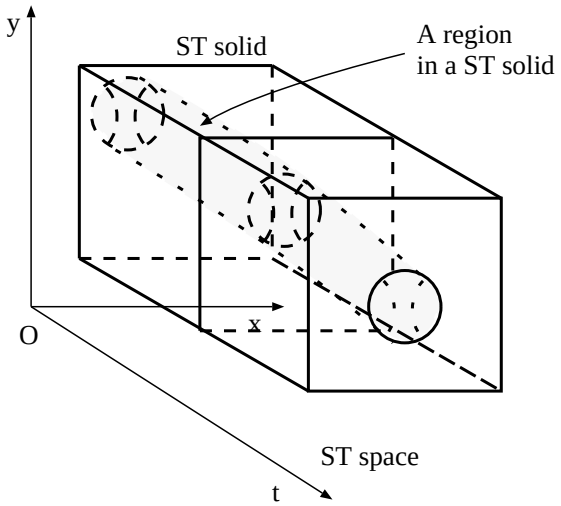


Figure 1: ST space and ST solid

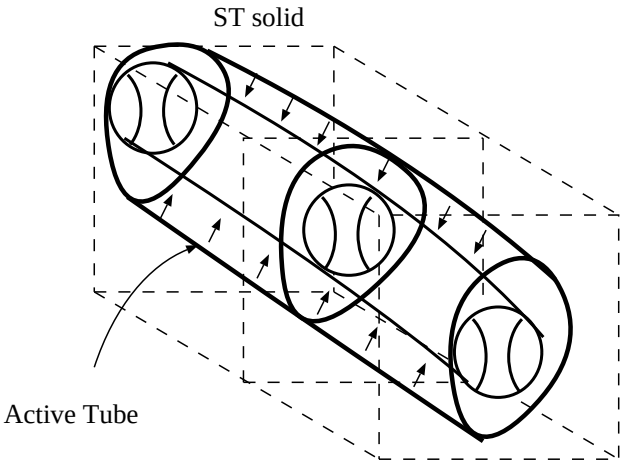


Figure 2: Active Tubes

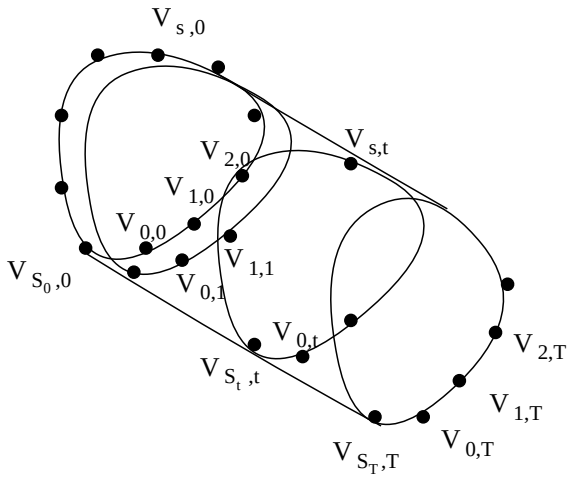


Figure 3: Structure of Active Tubes

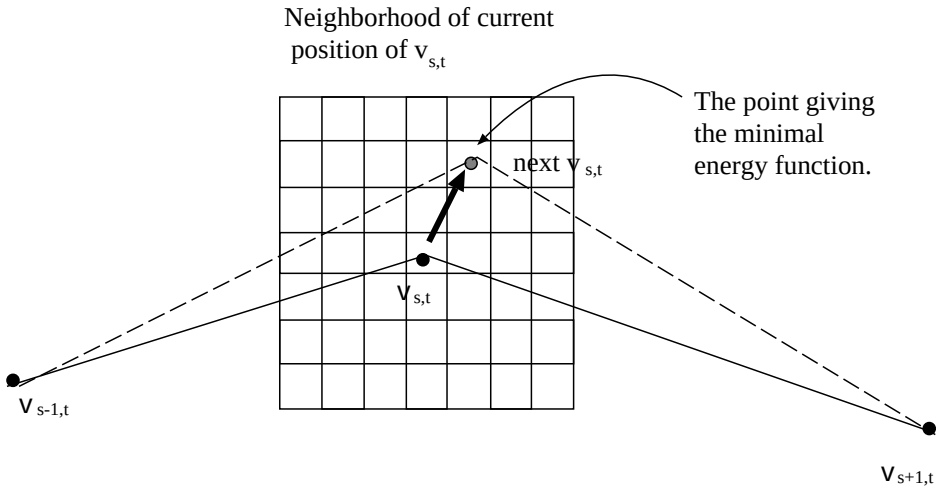


Figure 4: Greedy Algorithm

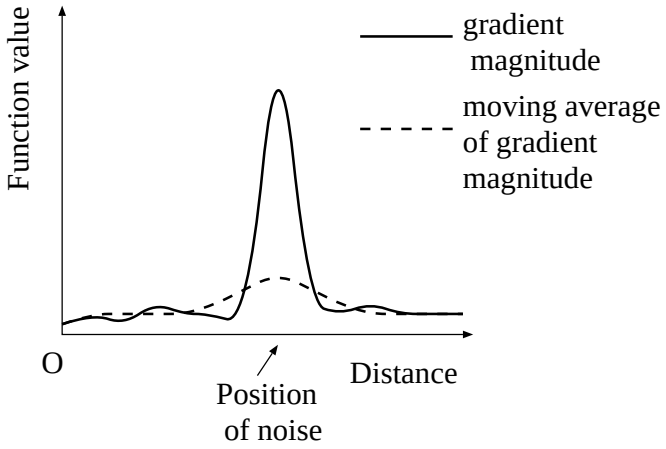


Figure 5: Effect of Noise (1): If the function of gradient magnitude is unimodal around a noise, using moving average or convolution can suppress the effect of the noise.

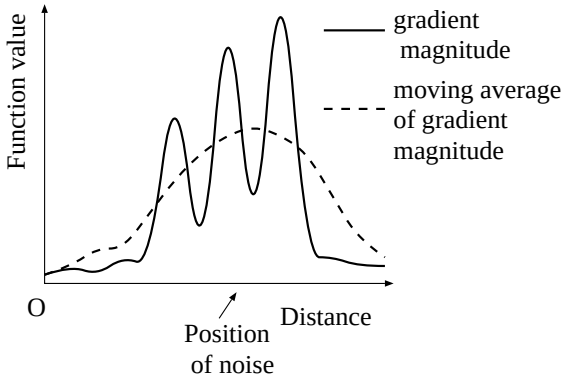


Figure 6: Effect of Noise (2): If several peaks of gradient magnitude are gathered around a noise, using moving average or convolution can't suppress the effect of the noise effectively.

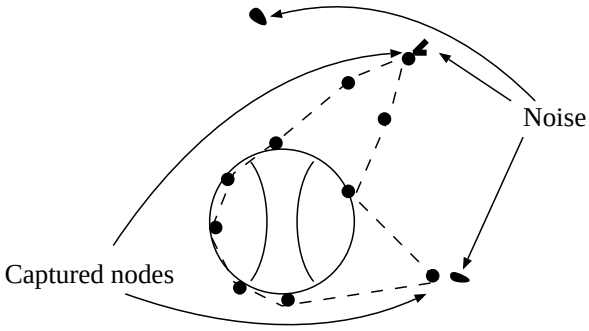


Figure 7: Capturing

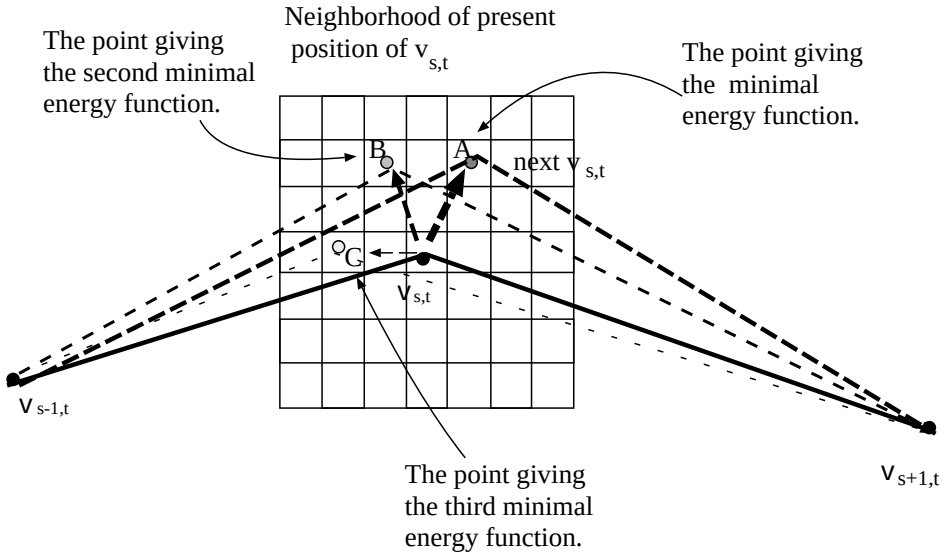


Figure 8: Randomized Greedy Algorithm

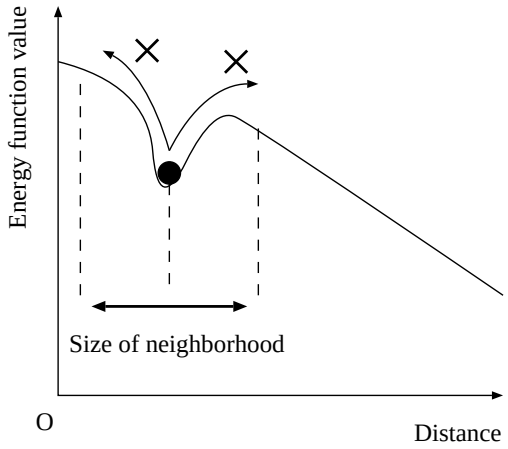


Figure 9: A Captured Node: Using Greedy Algorithm, a captured node can't escape from the pit.

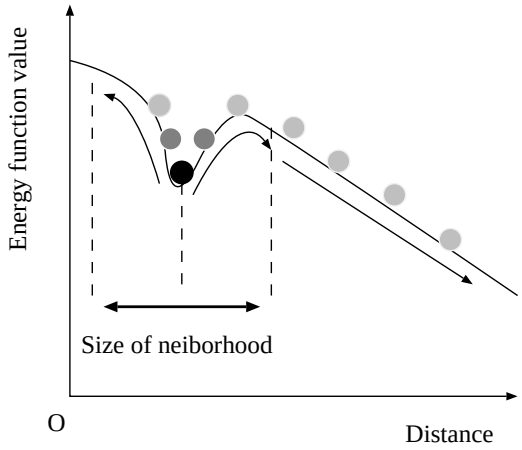


Figure 10: Avoiding Capturing: Using RGA, a captured node may be able to escape from the pit.

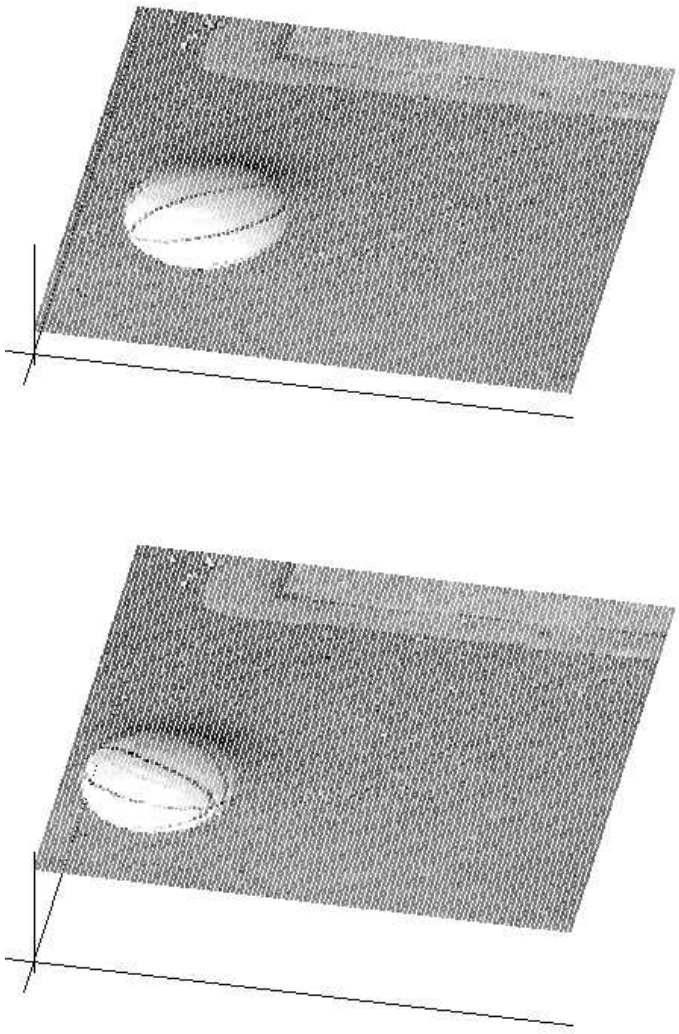


Figure 11: Sample Data for Experiments

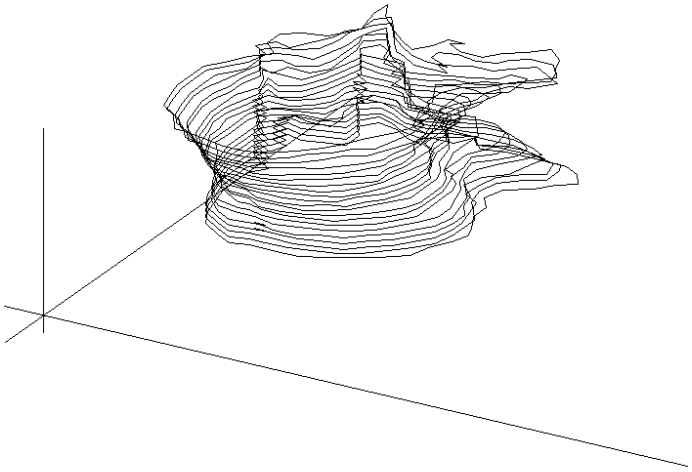
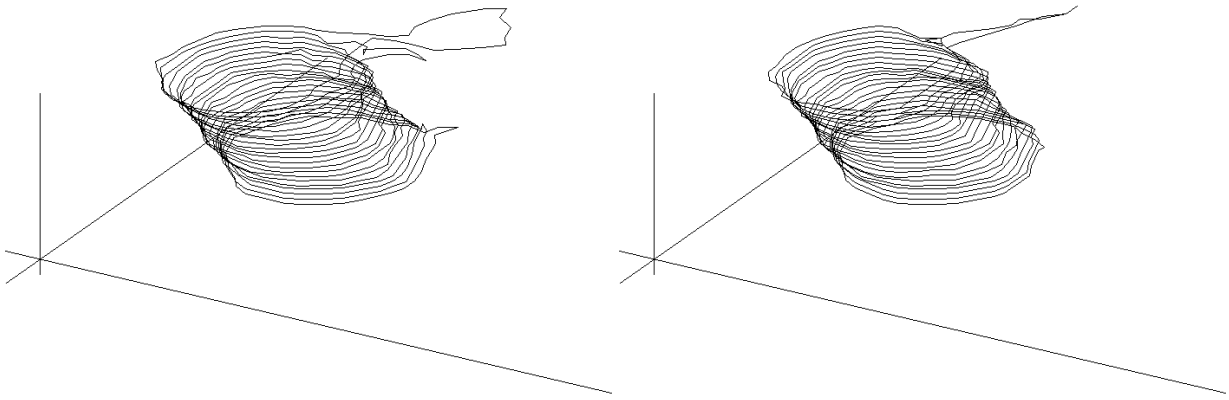
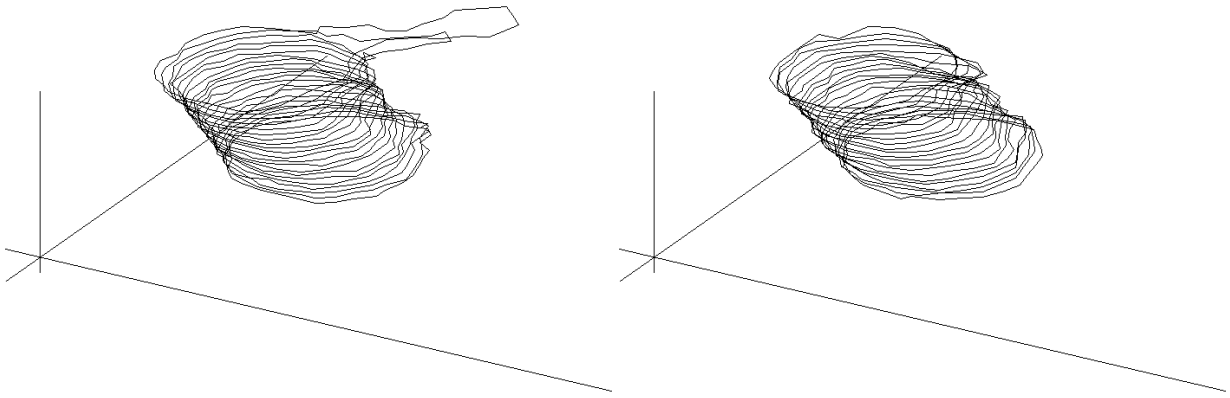


Figure 12: Initial Surface for Experiments



(a) (b)

Figure 13: Result of Greedy Algorithm, (a) After 10 iterations , (b) After convergence at 36th iteration



(a) (b)

Figure 14: Result of RGA, (a) After 10 iterations , (b) After convergence at 24th iteration