

Terrain Generation Assignment

Terrain Generation Assignment

Part 1: design

Design terrain generation system

Motivate *why* your approach is suitable

Sources!

Terrain Generation Assignment

Part 1: design

Design terrain generation system

Motivate *why* your approach is suitable

Sources!

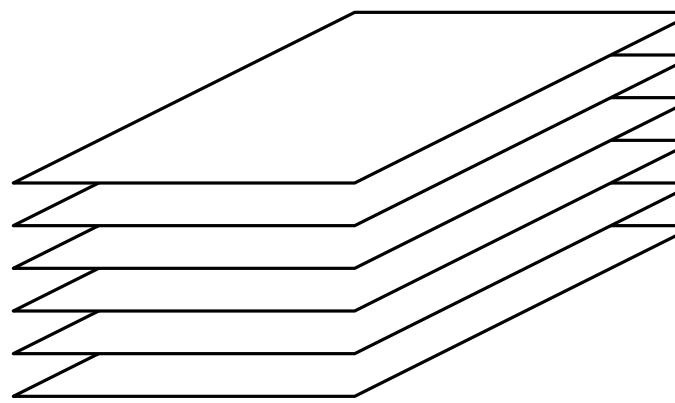
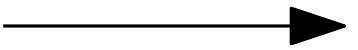
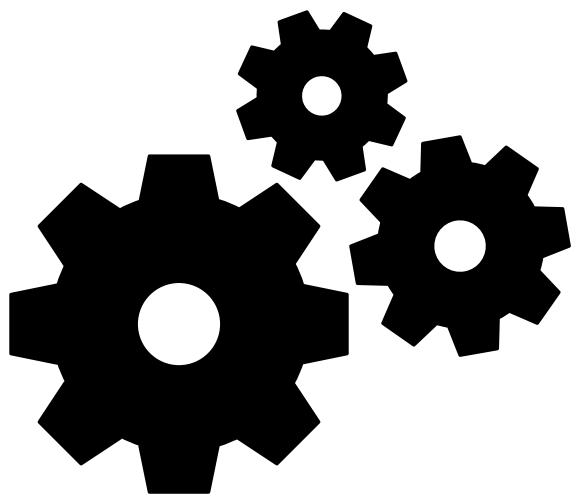
Part 2: implement

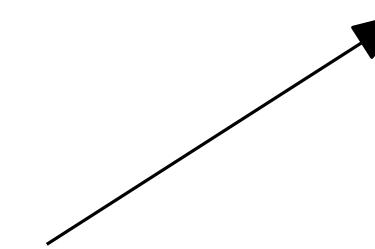
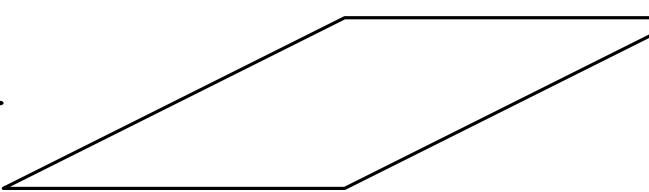
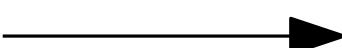
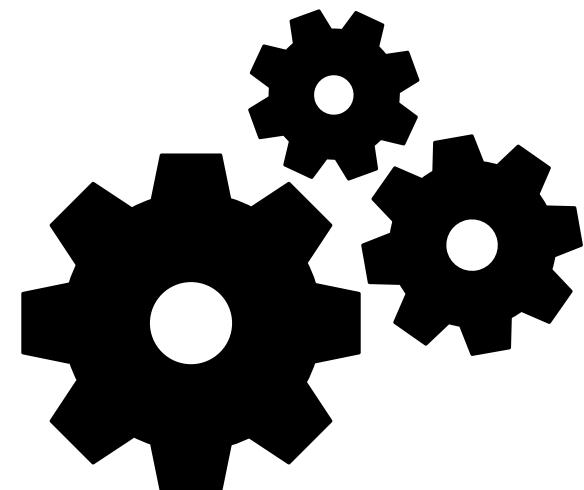
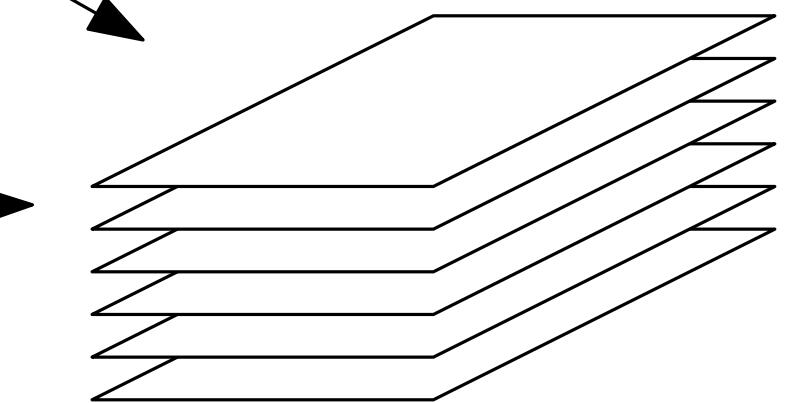
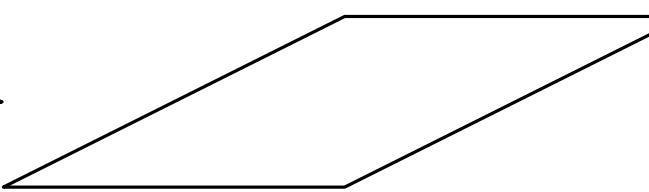
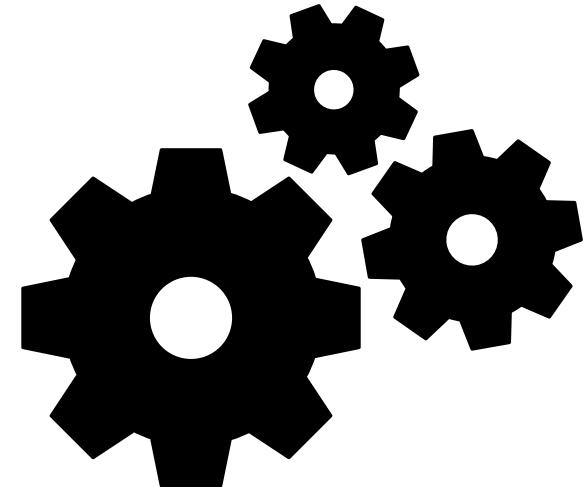
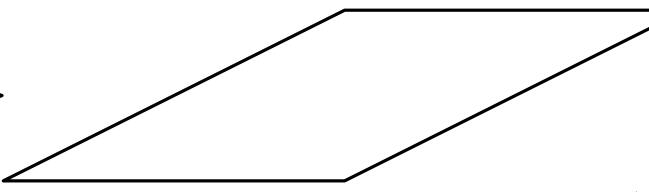
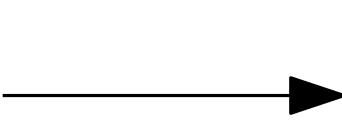
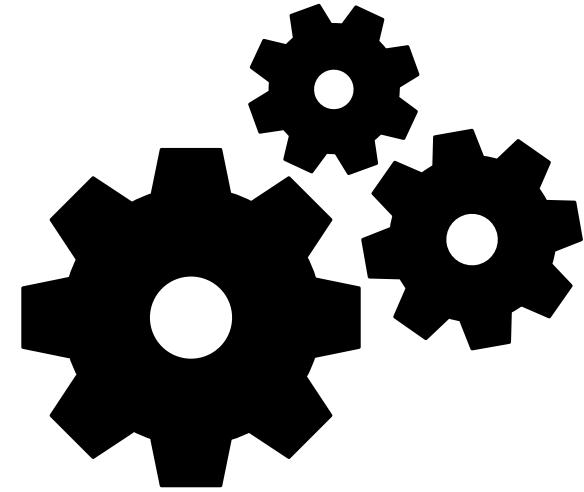
Implement terrain generation system

Explain differences with design

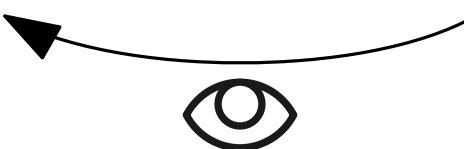
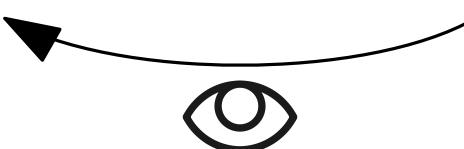
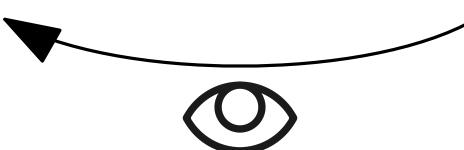
Agent-based PCG

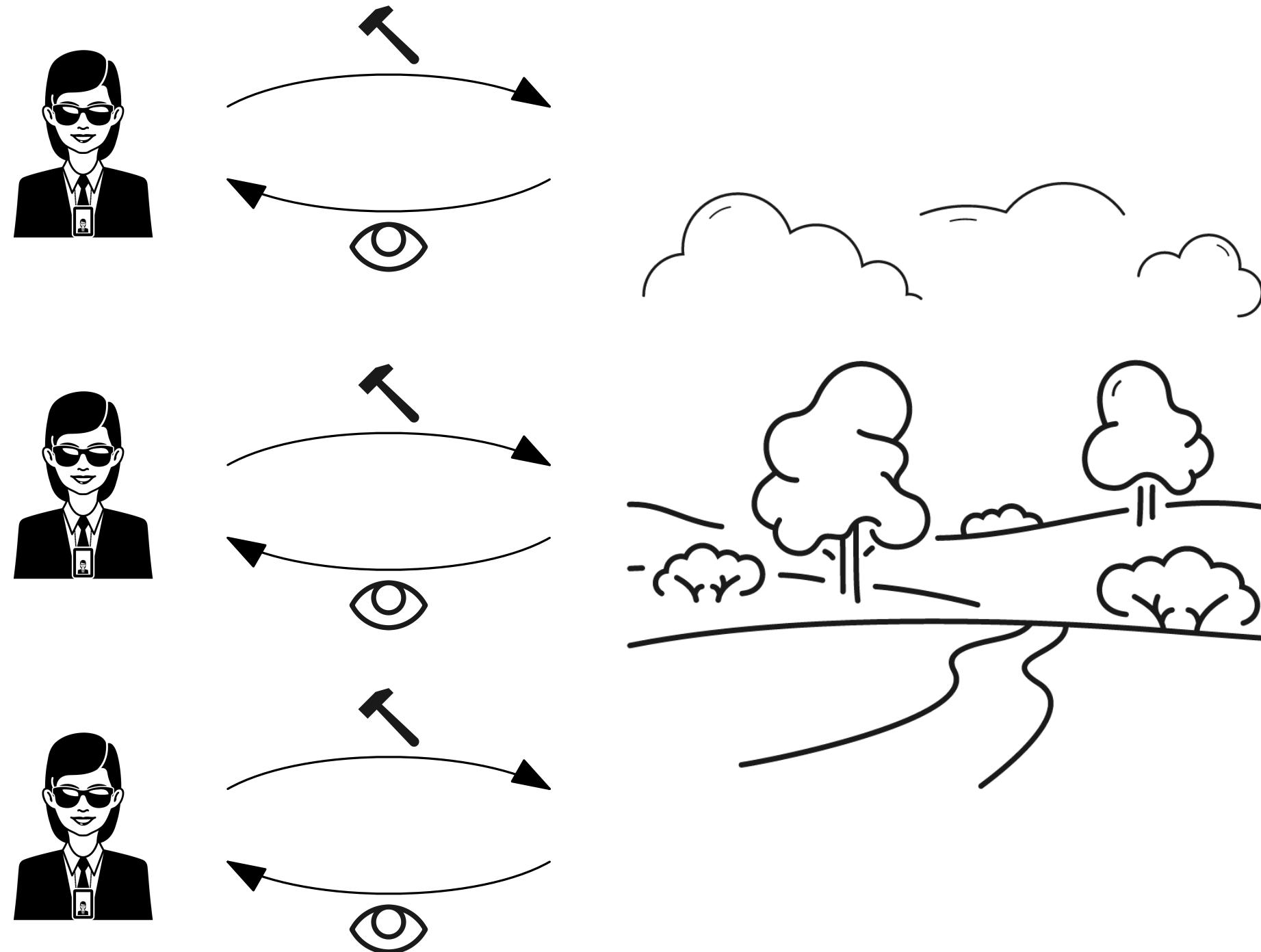
(And terrains)





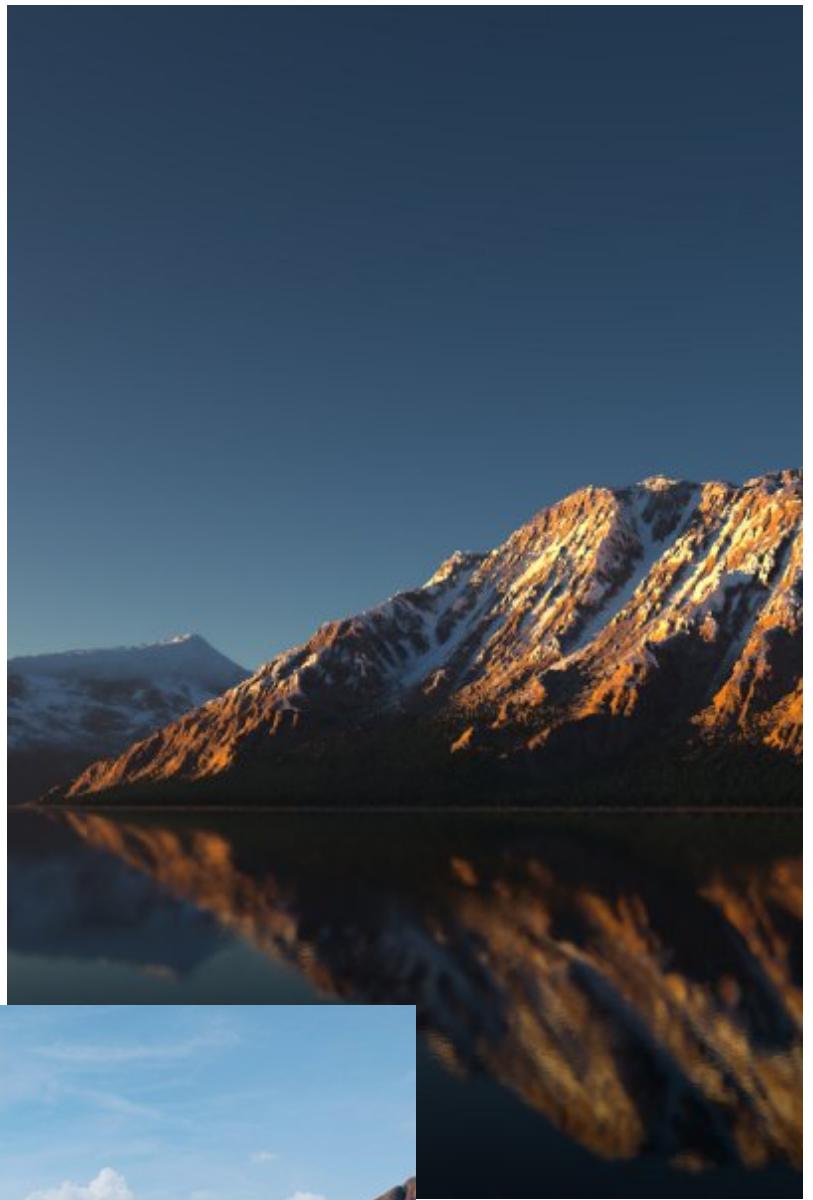






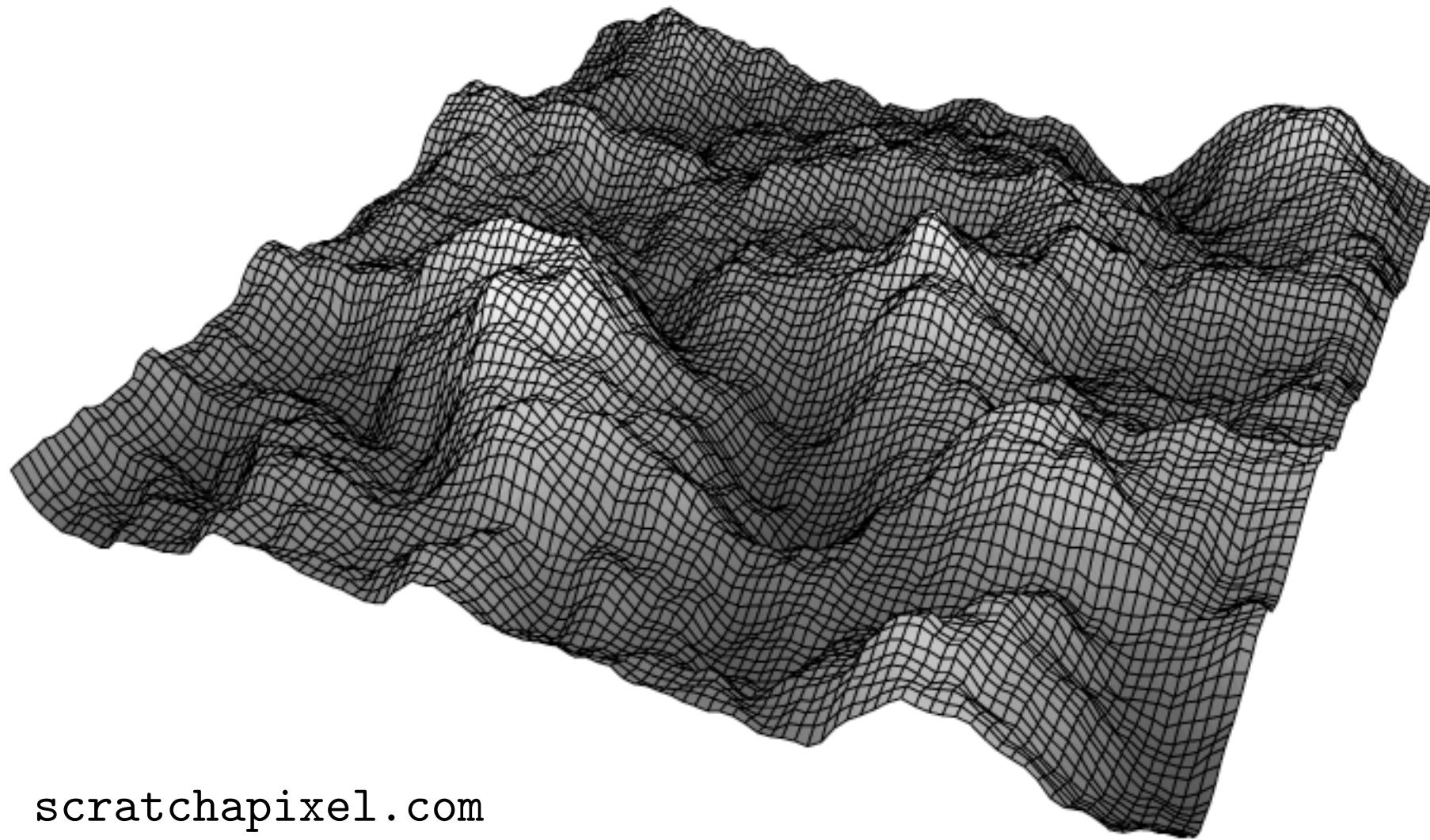


Terrain



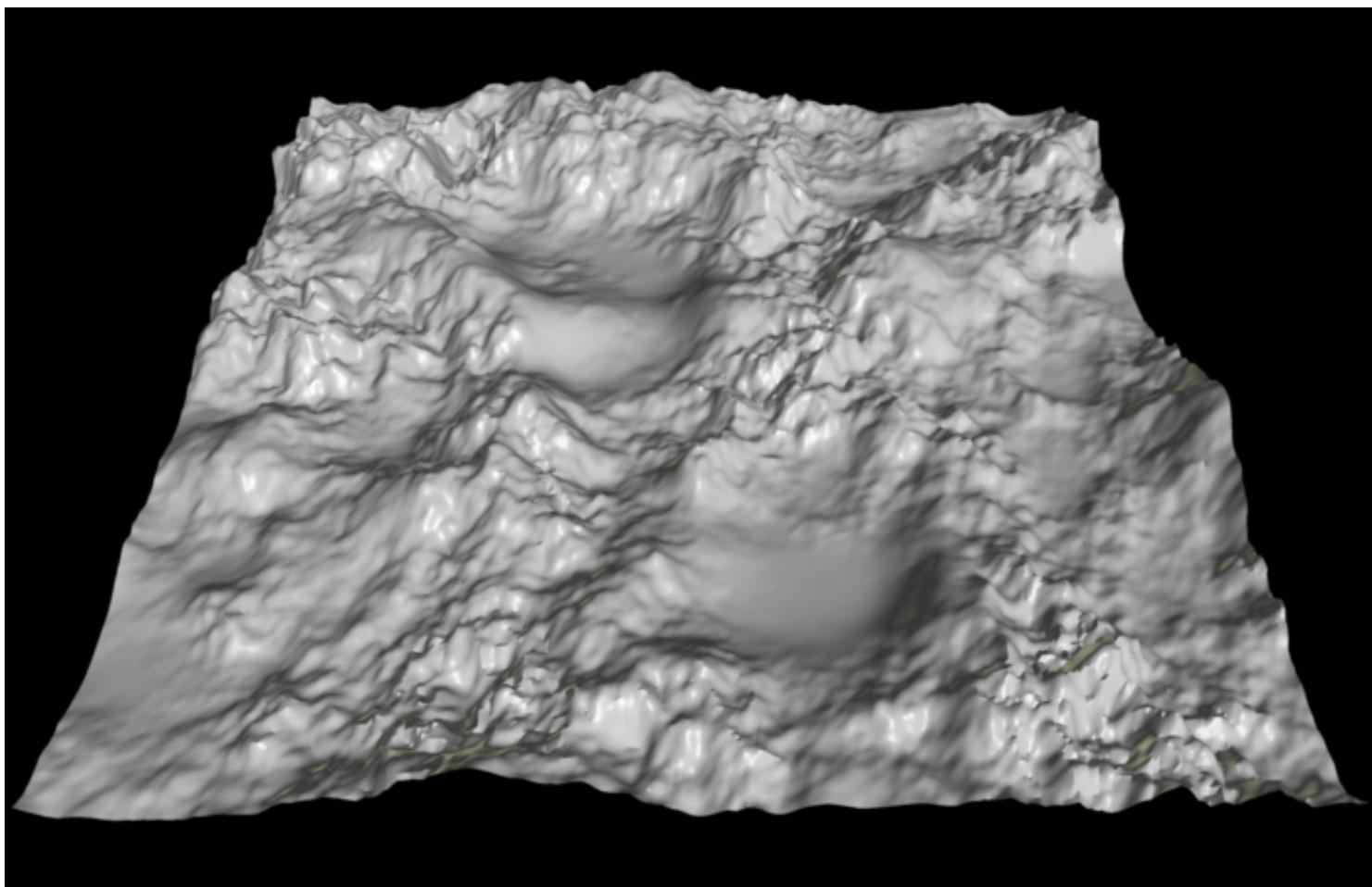
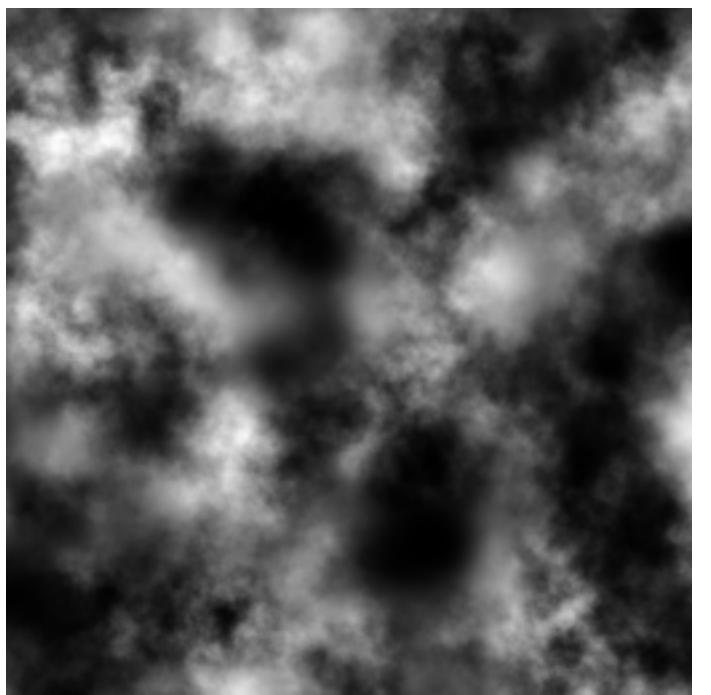
What is terrain?





scratchapixel.com



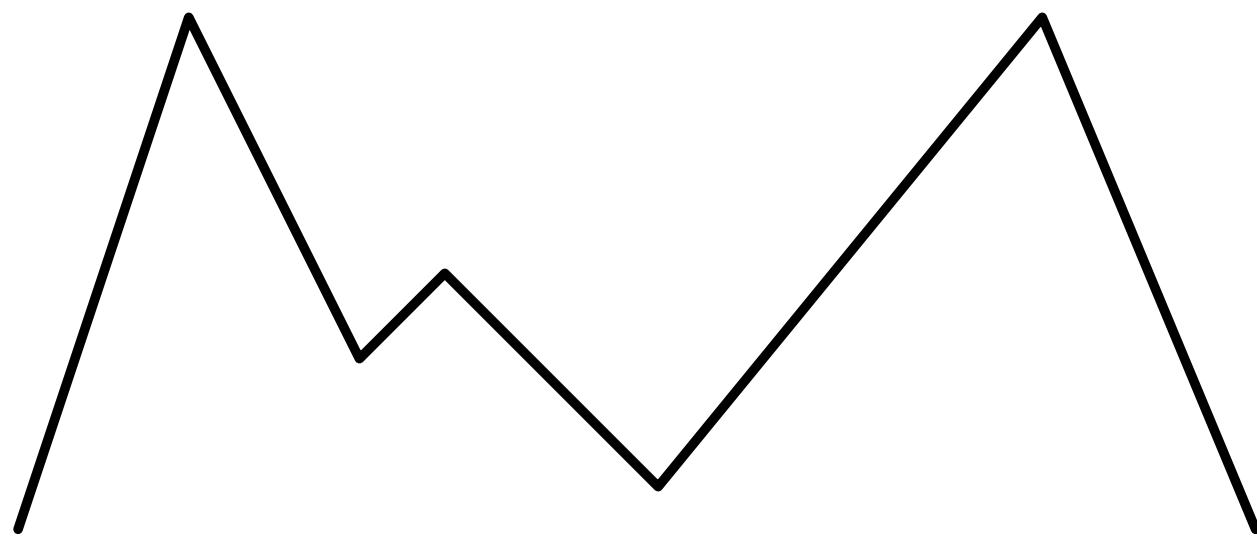


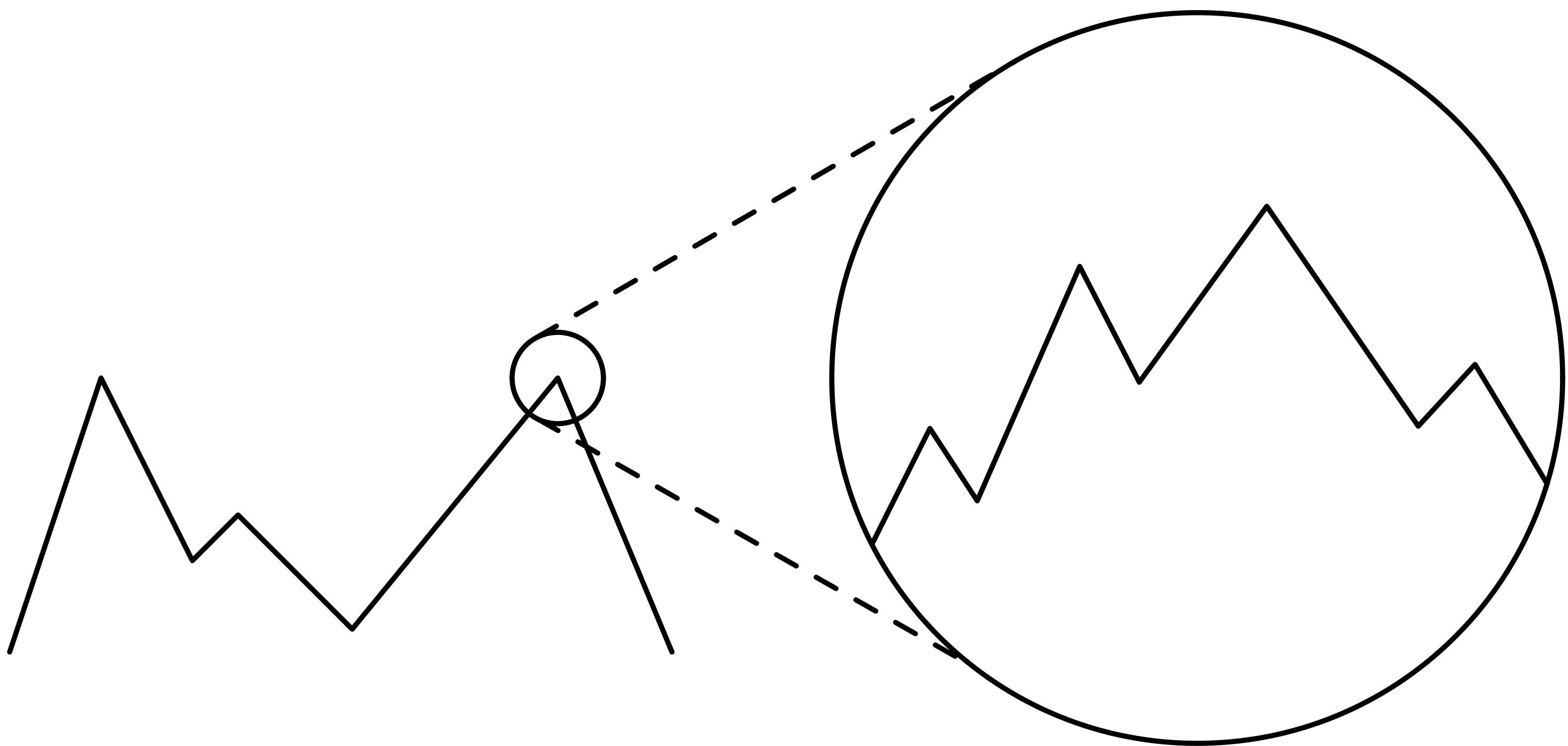
en.wikipedia.org/wiki/Heightmap

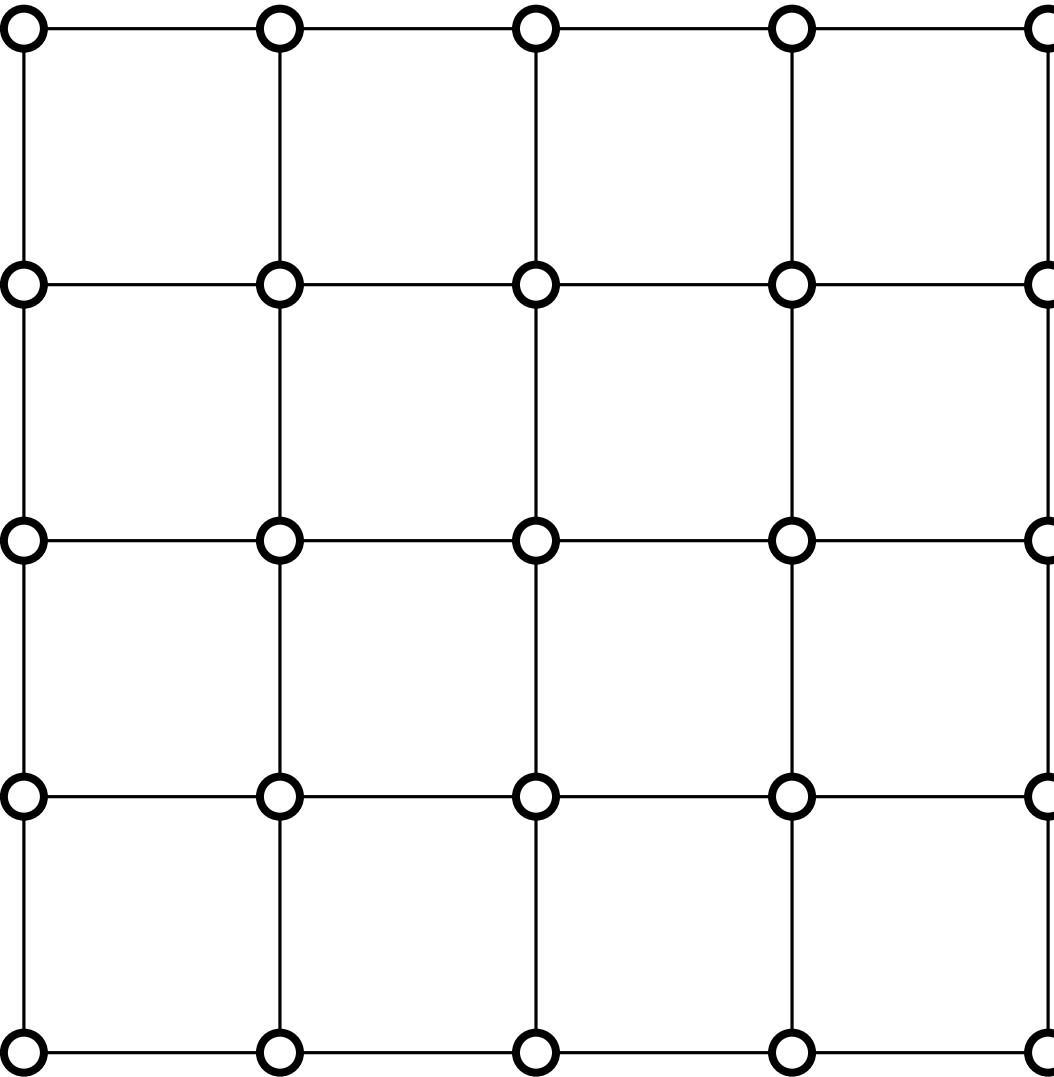
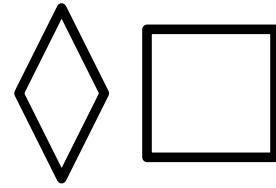
Assignment 1

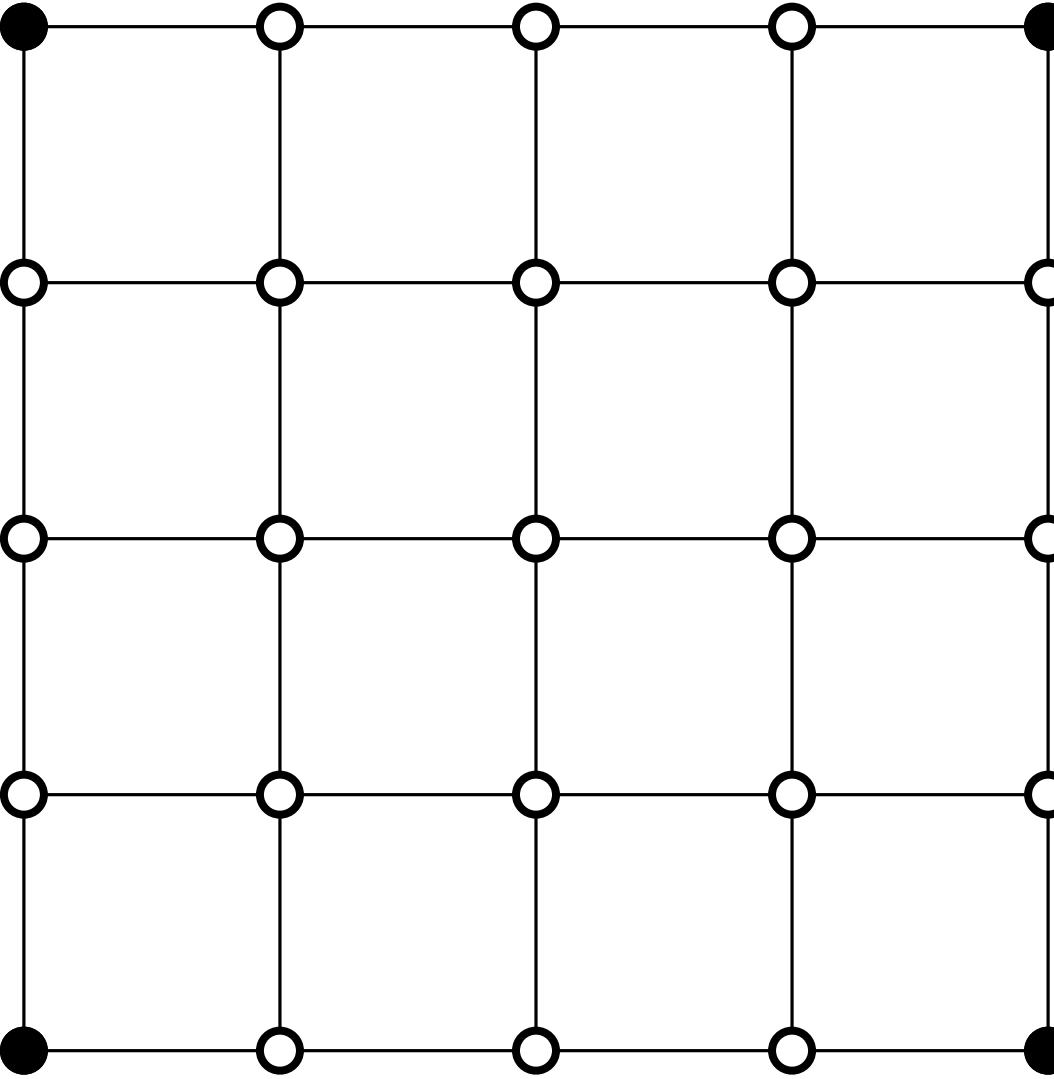
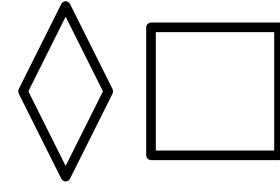
Name two disadvantages of representing terrain with a heightmap.

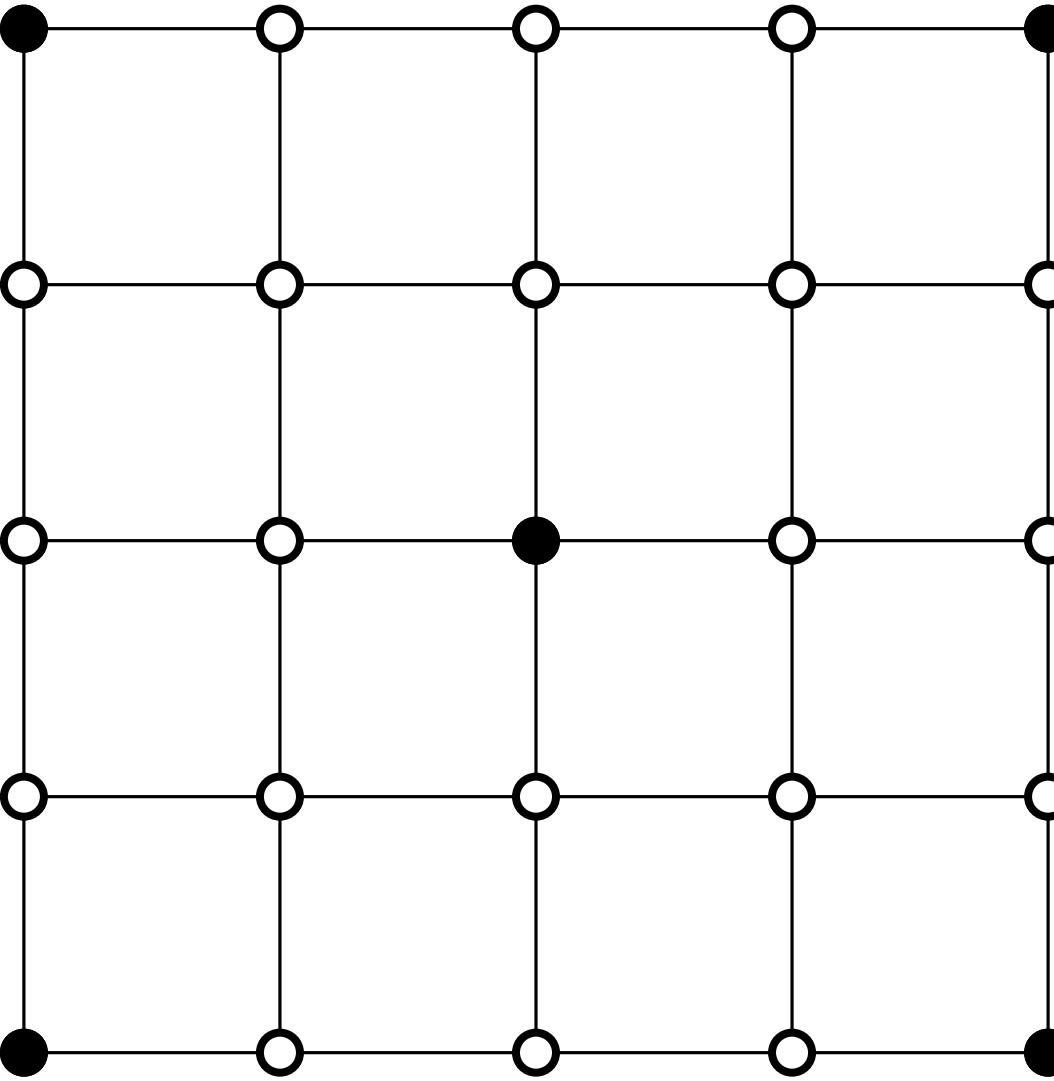
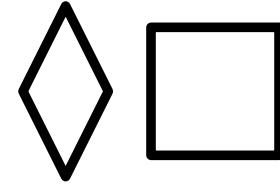
Fractal terrain

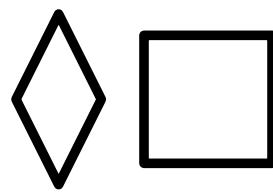
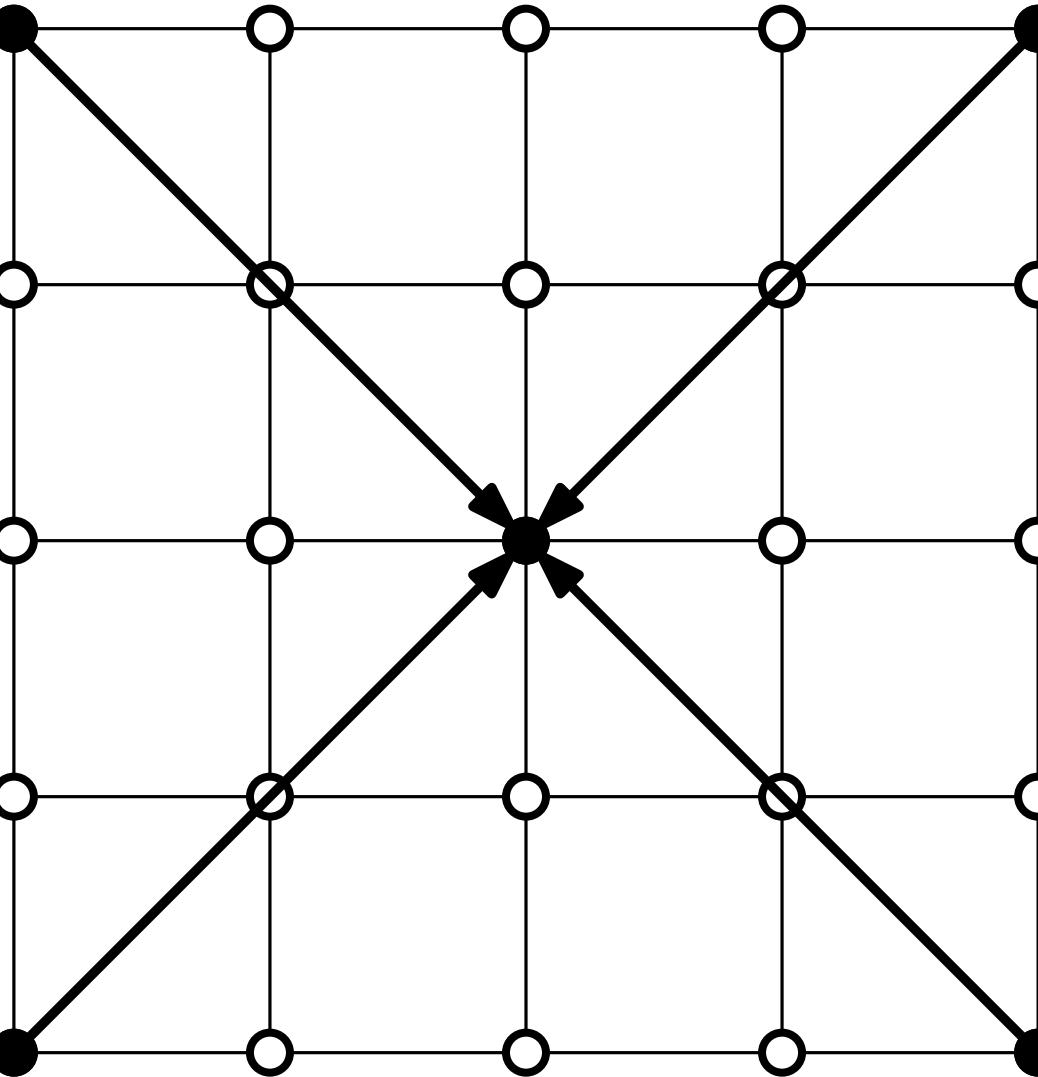


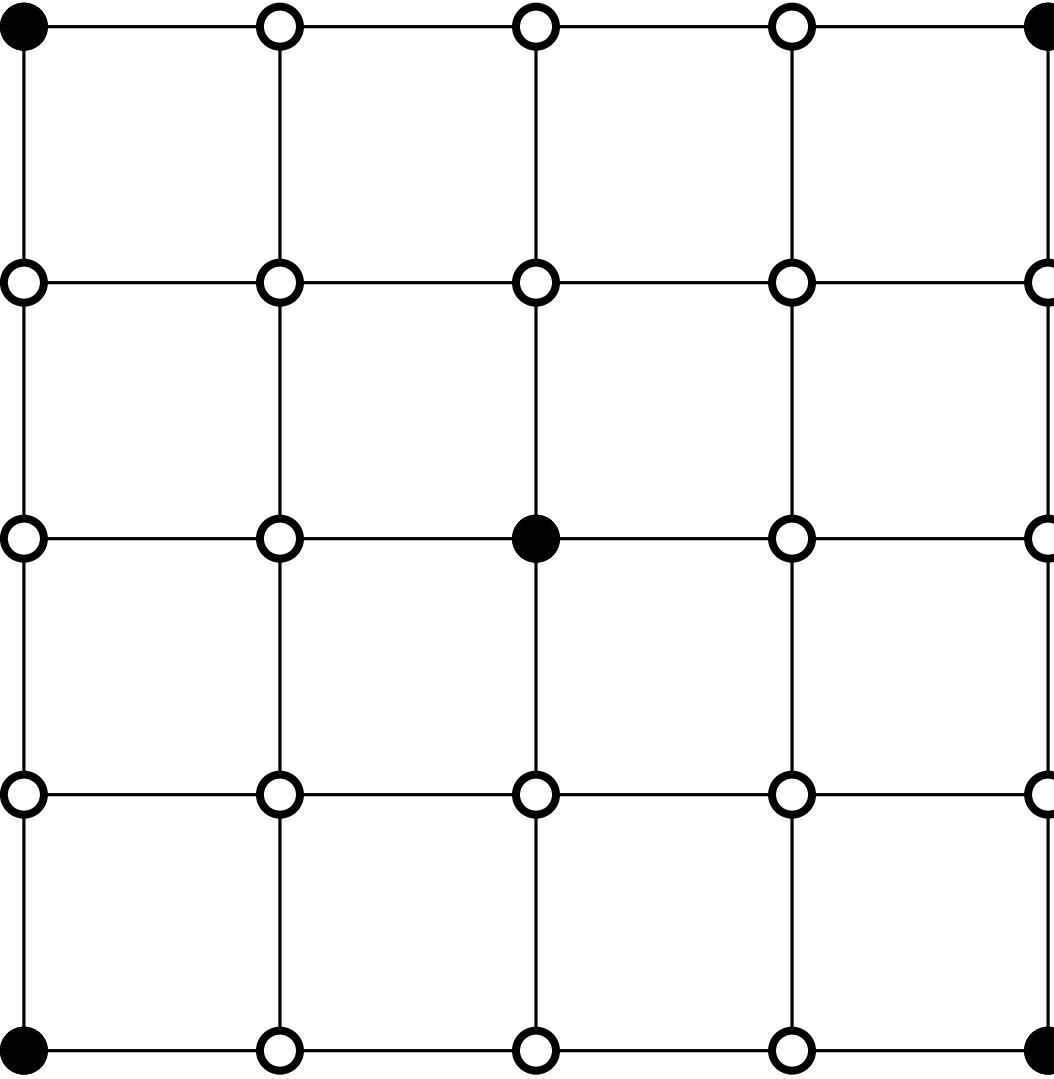
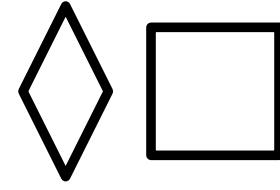


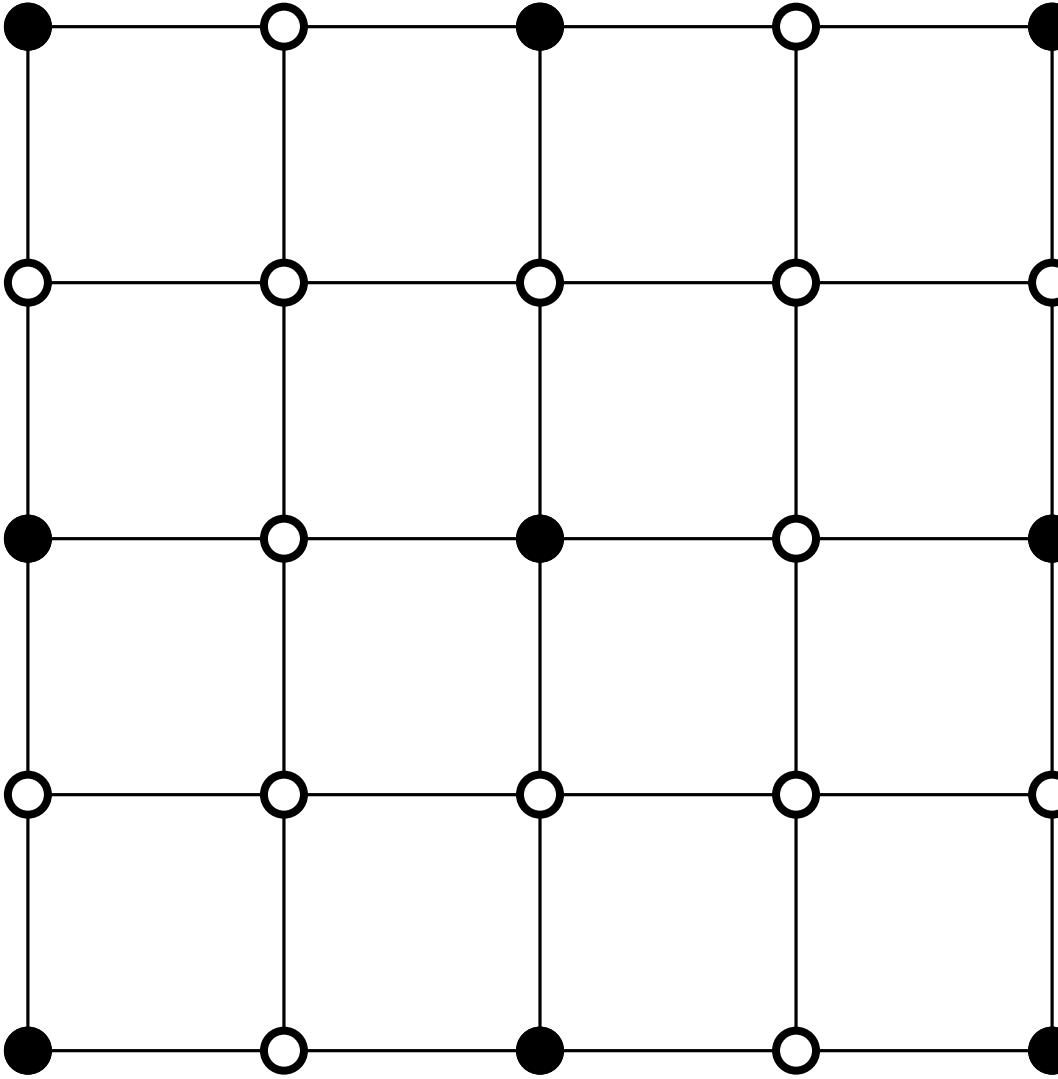
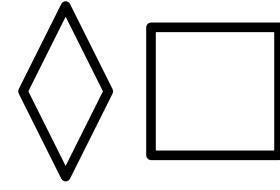


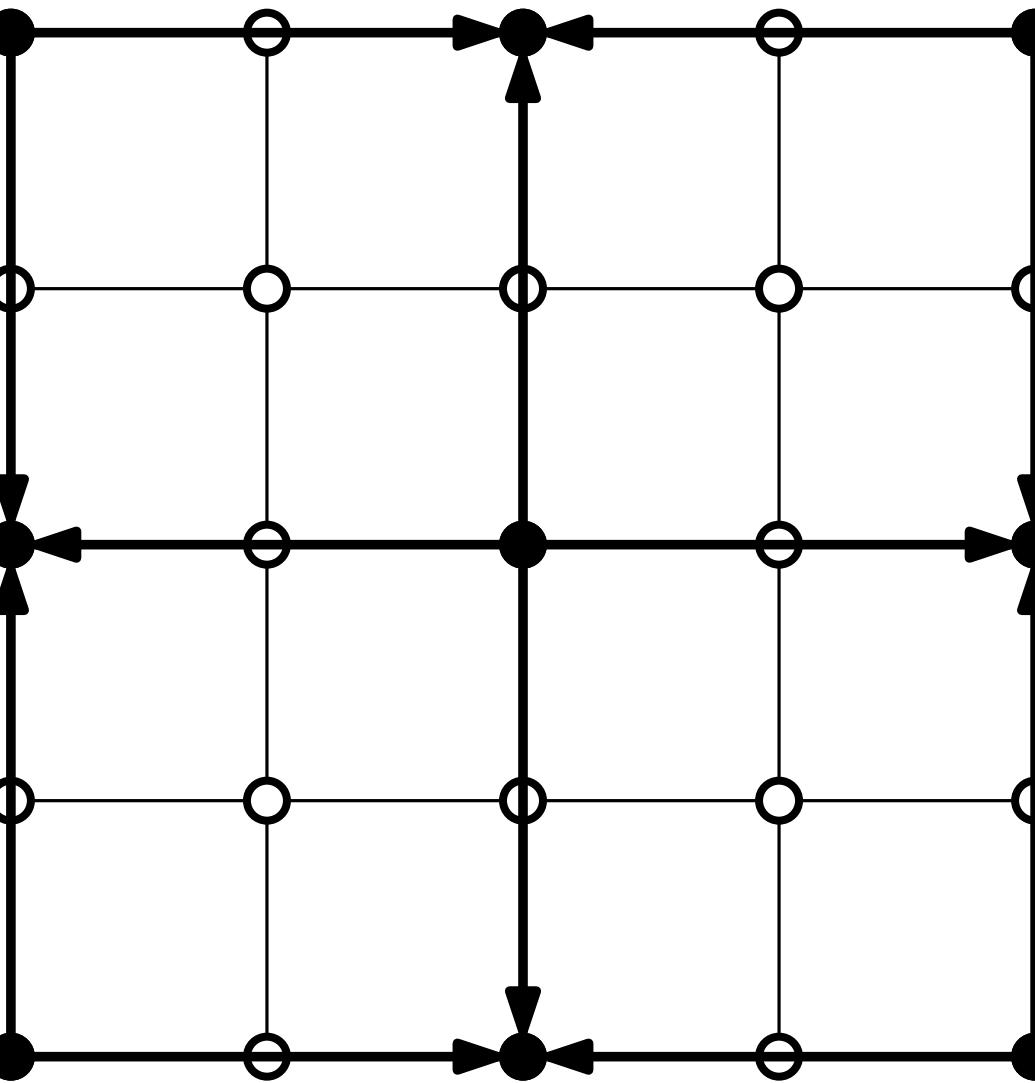
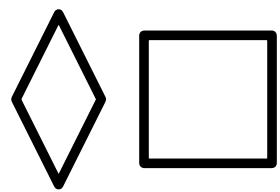




$$\frac{\Sigma}{n} +$$


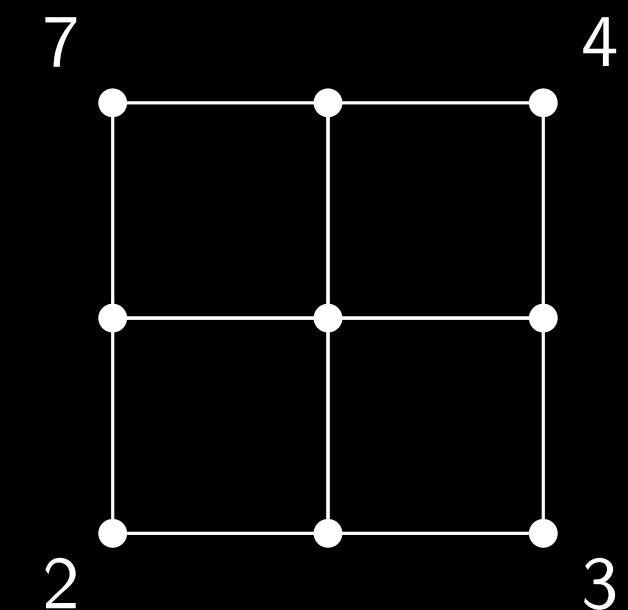




$$\frac{\Sigma}{n} +$$


Assignment 2

Assuming that the random number generator always returns 1,
complete the heightmap below using the diamond-square method.



Agent-based terrain

Controlled Procedural Terrain Generation Using Software Agents

Jonathon Doran*

Ian Parberry†

Dept. of Computer Science & Engineering
University of North Texas

January 19, 2010

Abstract

Procedural terrain generation is used to create landforms for applications such as computer games and flight simulators. While most of the existing work has concentrated on algorithms that generate terrain without input from the user, we explore a more controllable system that uses intelligent agents to generate terrain elevation heightmaps according to designer-defined constraints. This allows the designer to create procedural terrain that has specific properties.

Keywords: Procedural content generation, terrain, agent.

1 Introduction

Player demand for more content is increasing as games grow in complexity and scope. Content such as terrain plays a fundamental role in certain types of games (Forbus [4]), and contributes greatly to replayability (Sampath [21]). Traditionally developers have hand crafted content for games, requiring a substantial investment of time and resources. Procedural content generation is the automated generation of these assets for games (Nelson and Mateas [13]). The desire for automatic terrain generation stems from the goal of providing the player with novel content without a large investment of developer resources. Prior work has shown that this generation is feasible (see, for example, [6, 14, 27]).

We feel that procedurally generated content should ideally have these properties:

- Novelty: contains an element of randomness and unpredictability.
- Structure: is not merely random noise, but contains larger structures.
- Interest: has a combination of randomness and structure that players find engaging.
- Speed: can be quickly generated.
- Controllability: can be generated according to a set of natural designer-centric parameters.

With this in mind, we present a terrain generator based on software agents. This generator creates heightmaps that are grids containing elevation data at regular points on the map. While our generator could generate arbitrary size maps, the examples presented in this paper were all

*Author's address: Dept. of Computer Science & Engineering, Univ. of North Texas, 1155 Union Circle #311366, Denton, Texas 76203-5017, U.S.A. Email: jhd@unt.edu.

†Email: ian@unt.edu.



en.wikipedia.org/wiki/Coast



en.wikipedia.org/wiki/Coast



en.wikipedia.org/wiki/Beach



en.wikipedia.org/wiki/Coast



en.wikipedia.org/wiki/Mountain



en.wikipedia.org/wiki/Beach



en.wikipedia.org/wiki/Coast



en.wikipedia.org/wiki/Mountain



en.wikipedia.org/wiki/Beach



en.wikipedia.org/wiki/River



en.wikipedia.org/wiki/Coast



en.wikipedia.org/wiki/Mountain



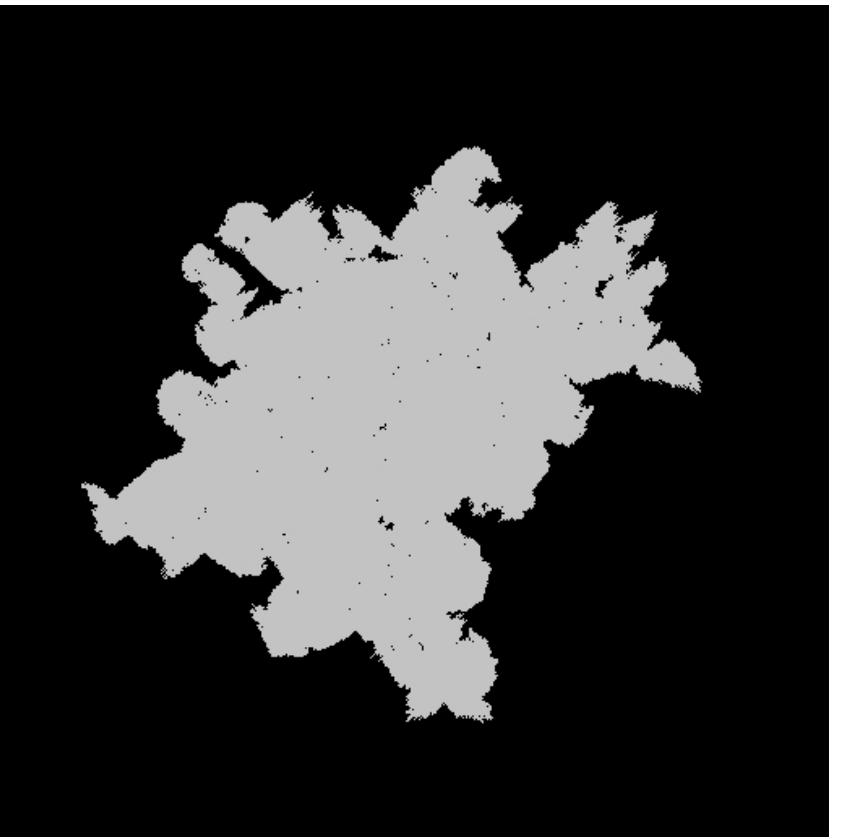
allposters.com

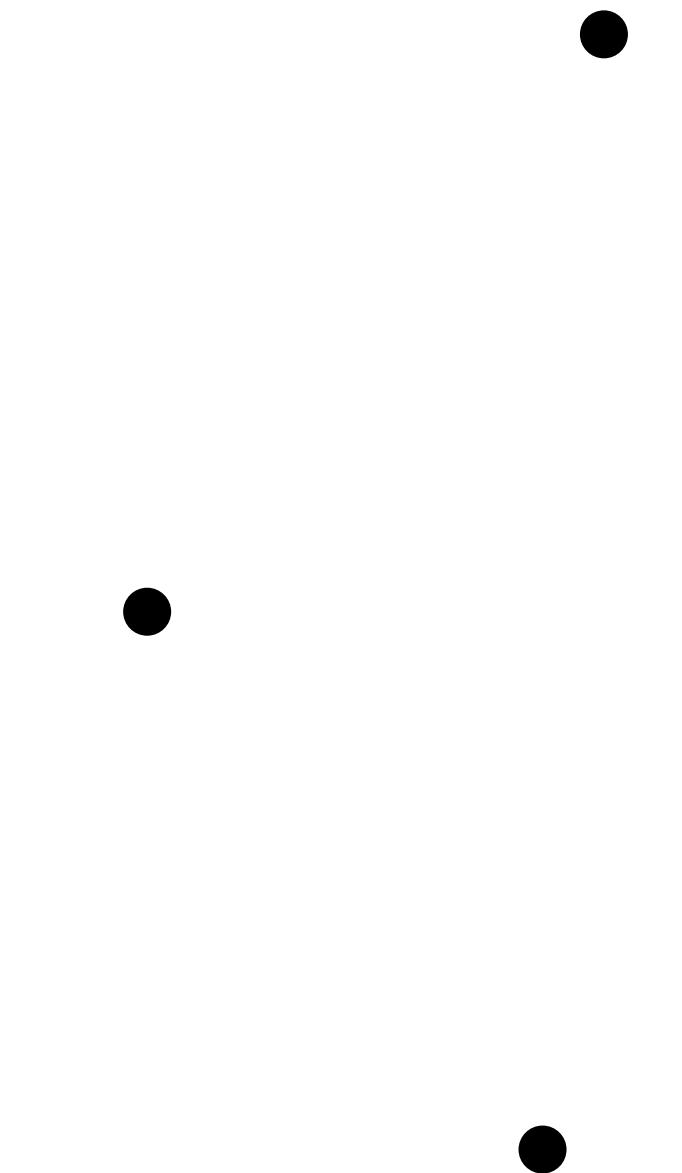
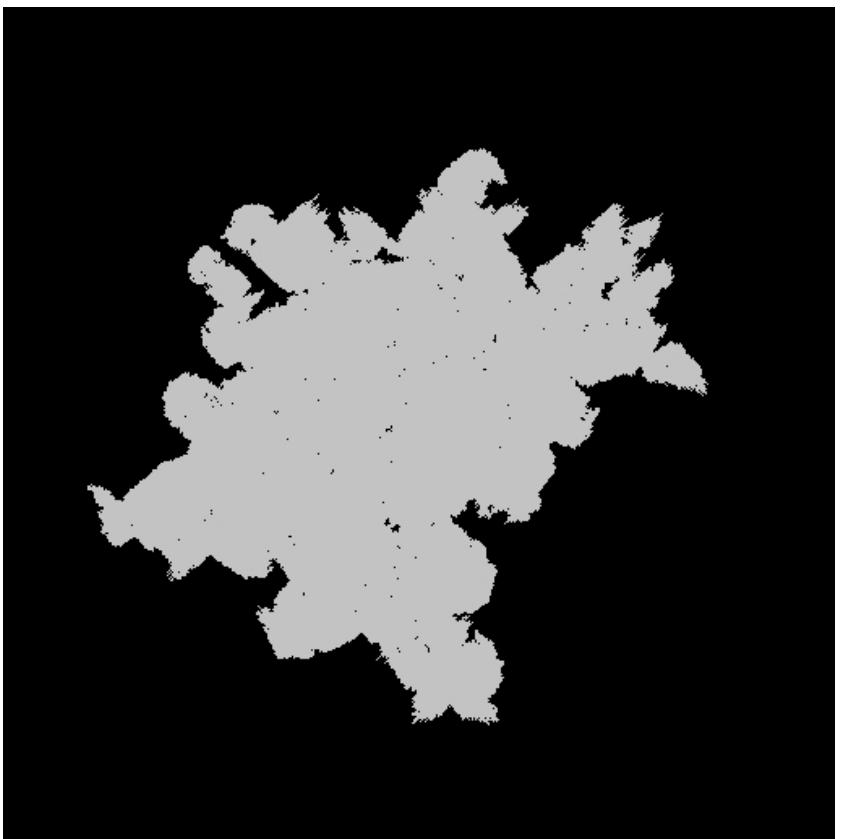


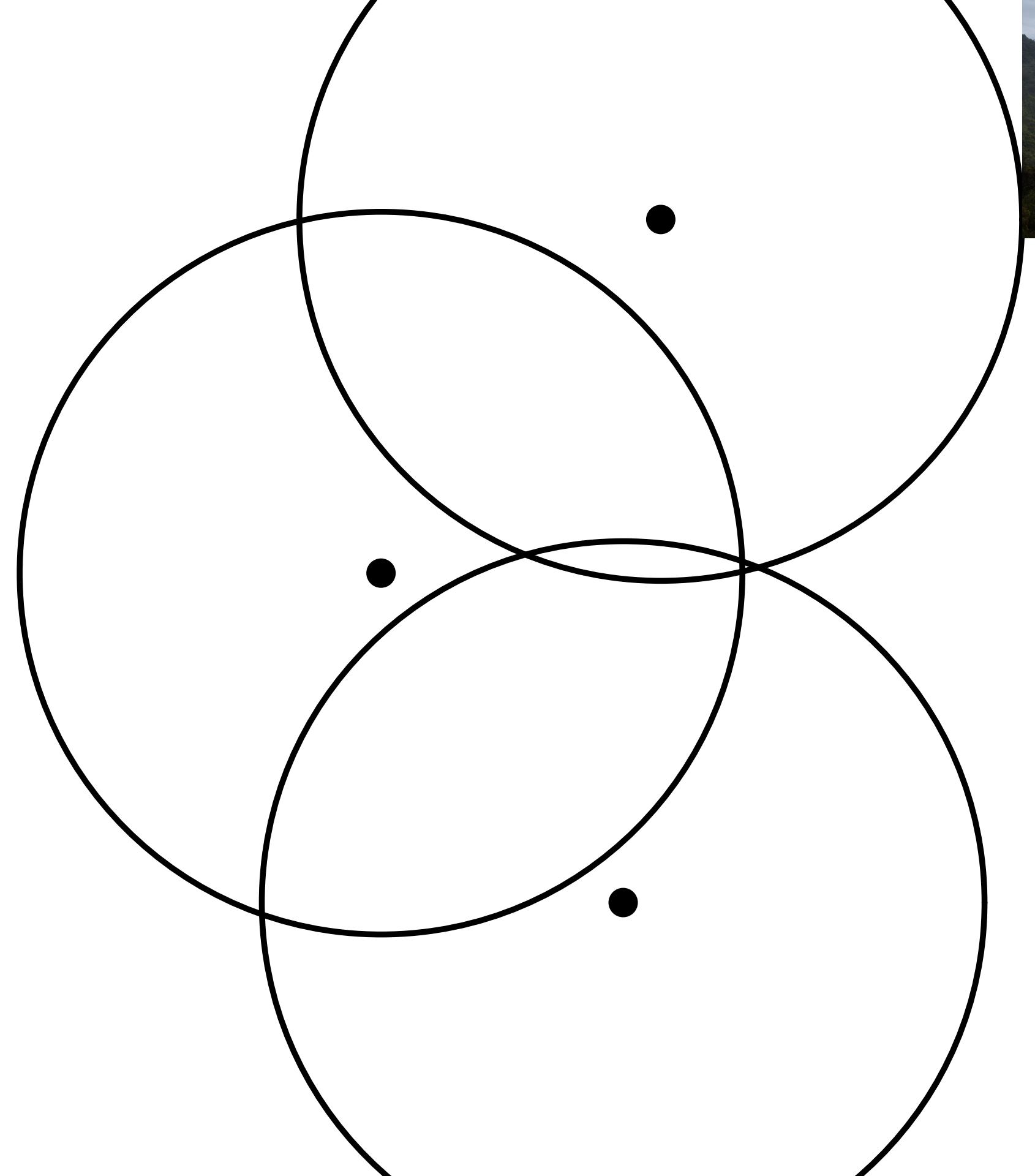
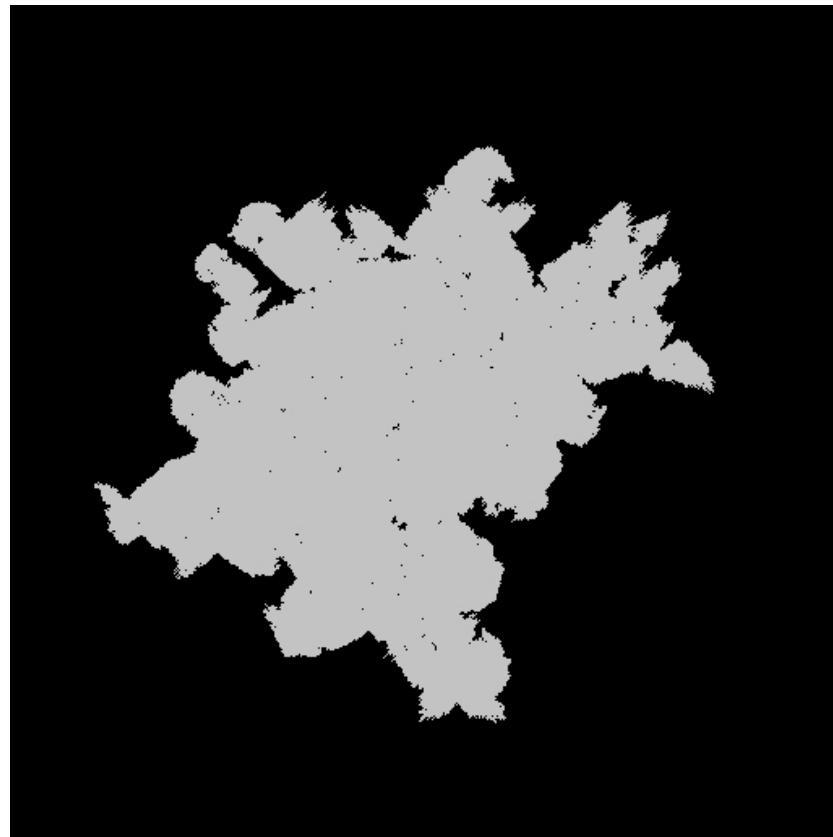
en.wikipedia.org/wiki/Beach

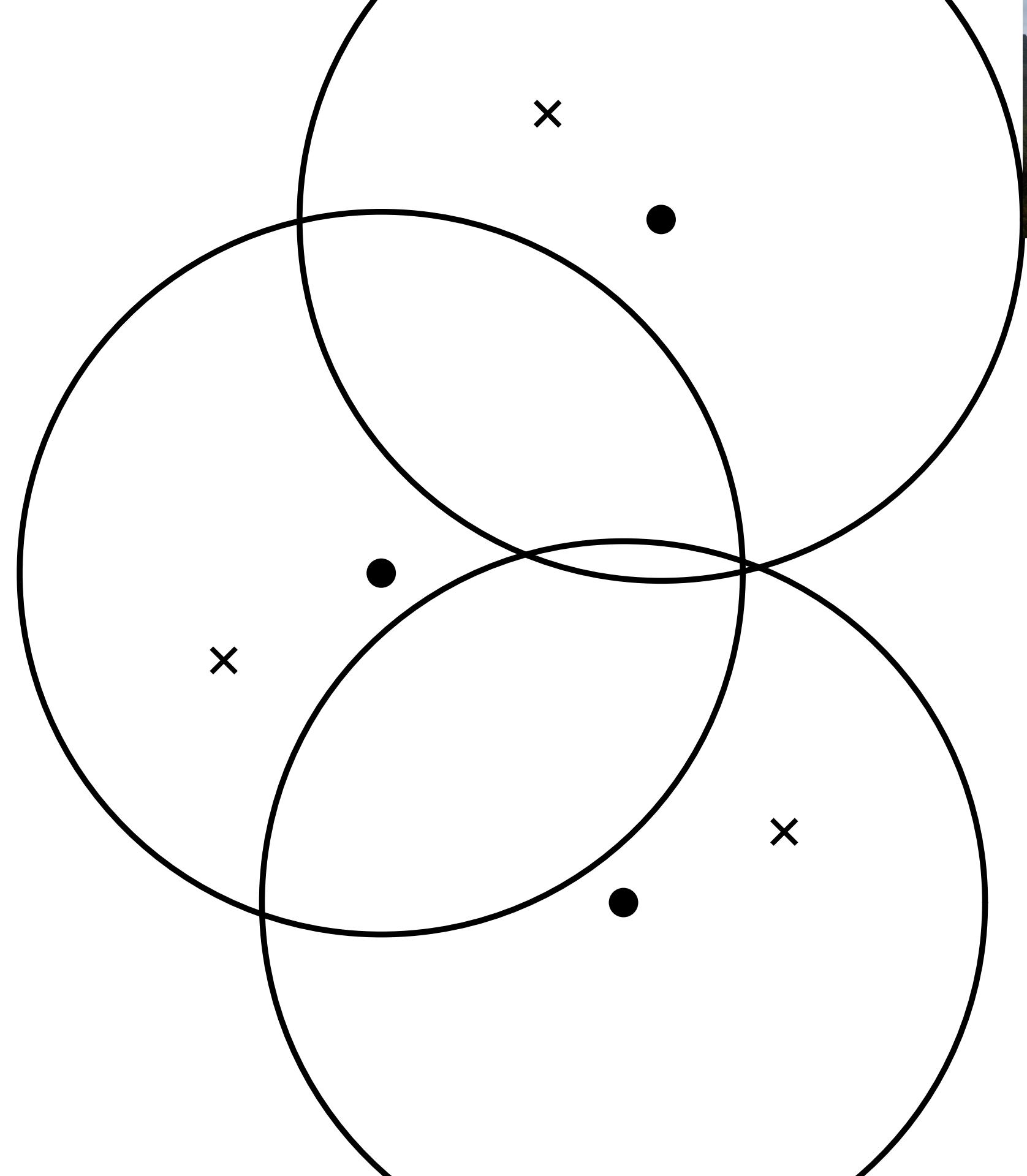
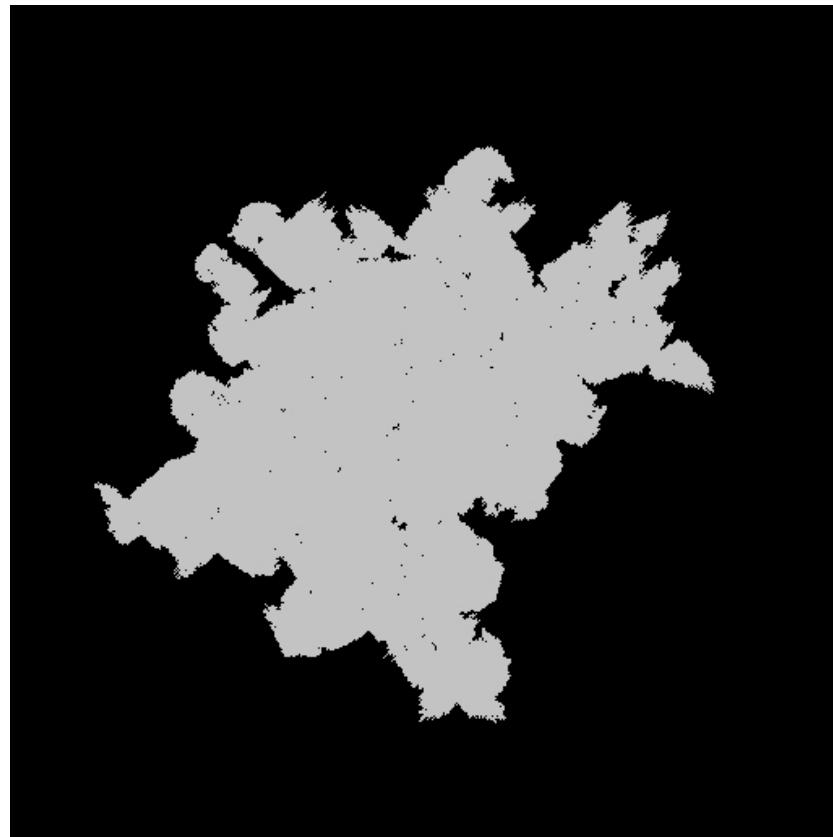


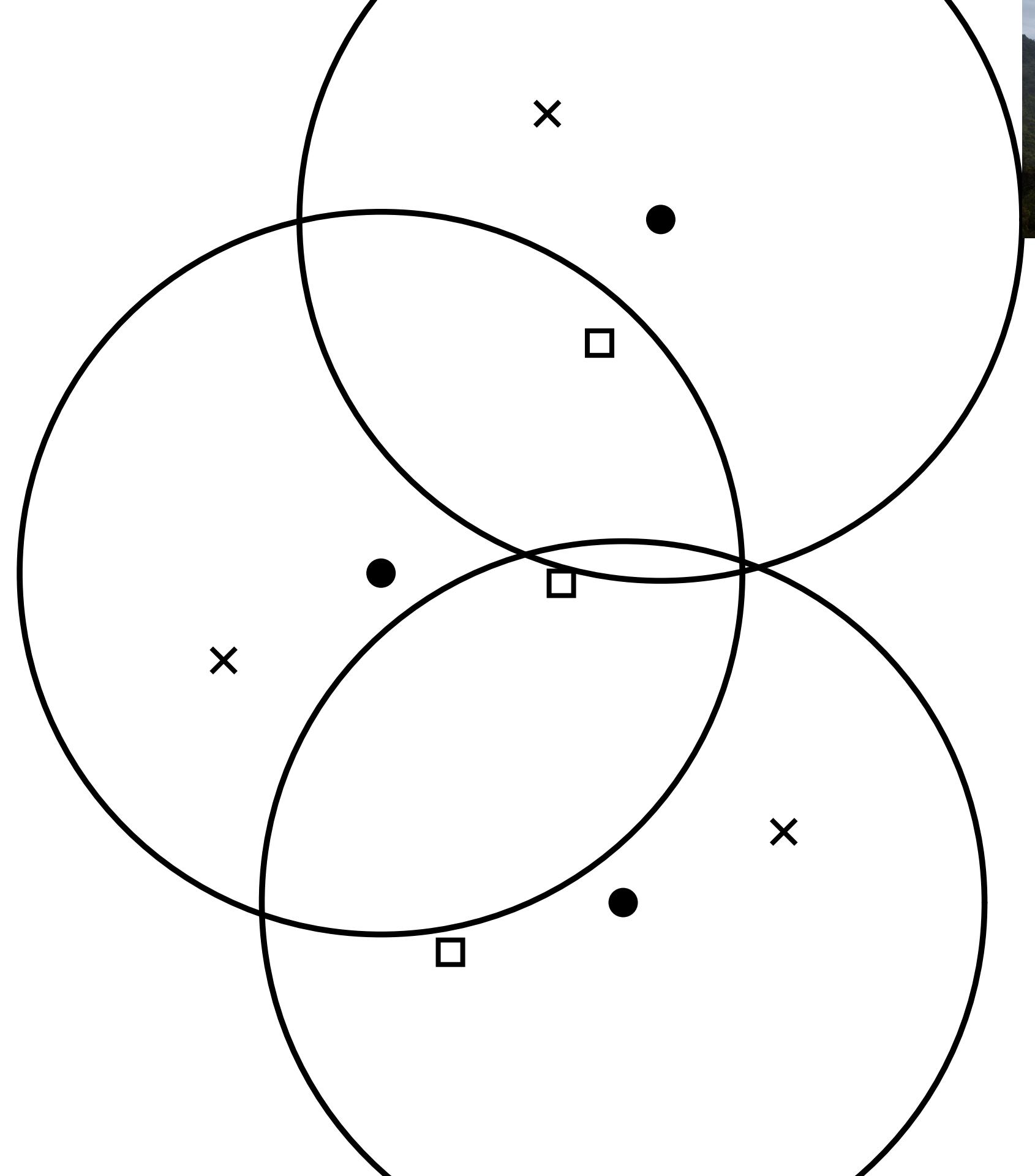
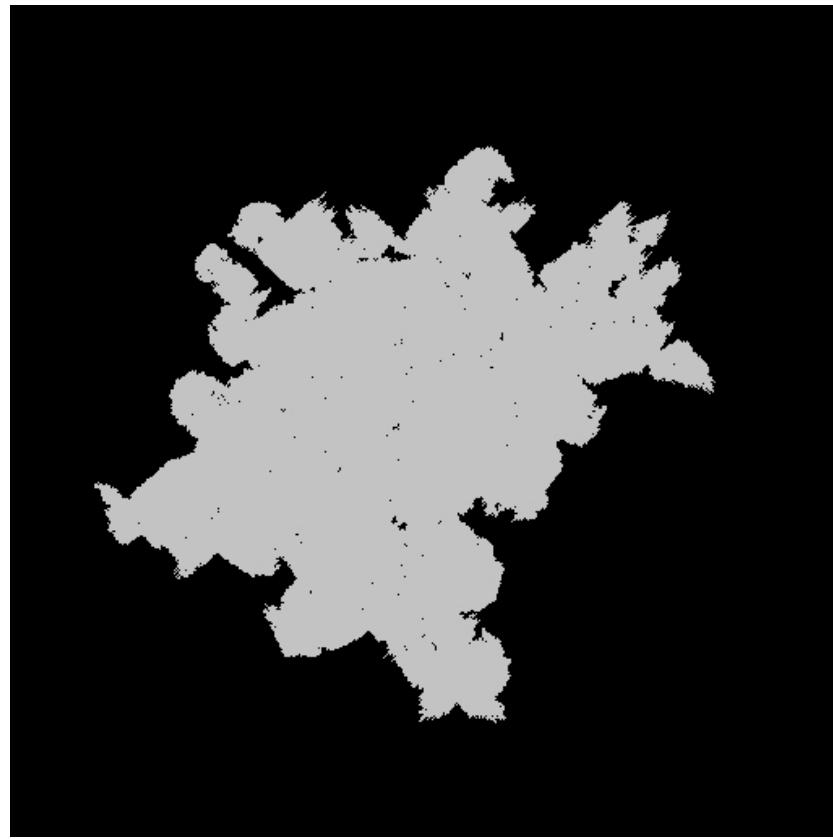
en.wikipedia.org/wiki/River

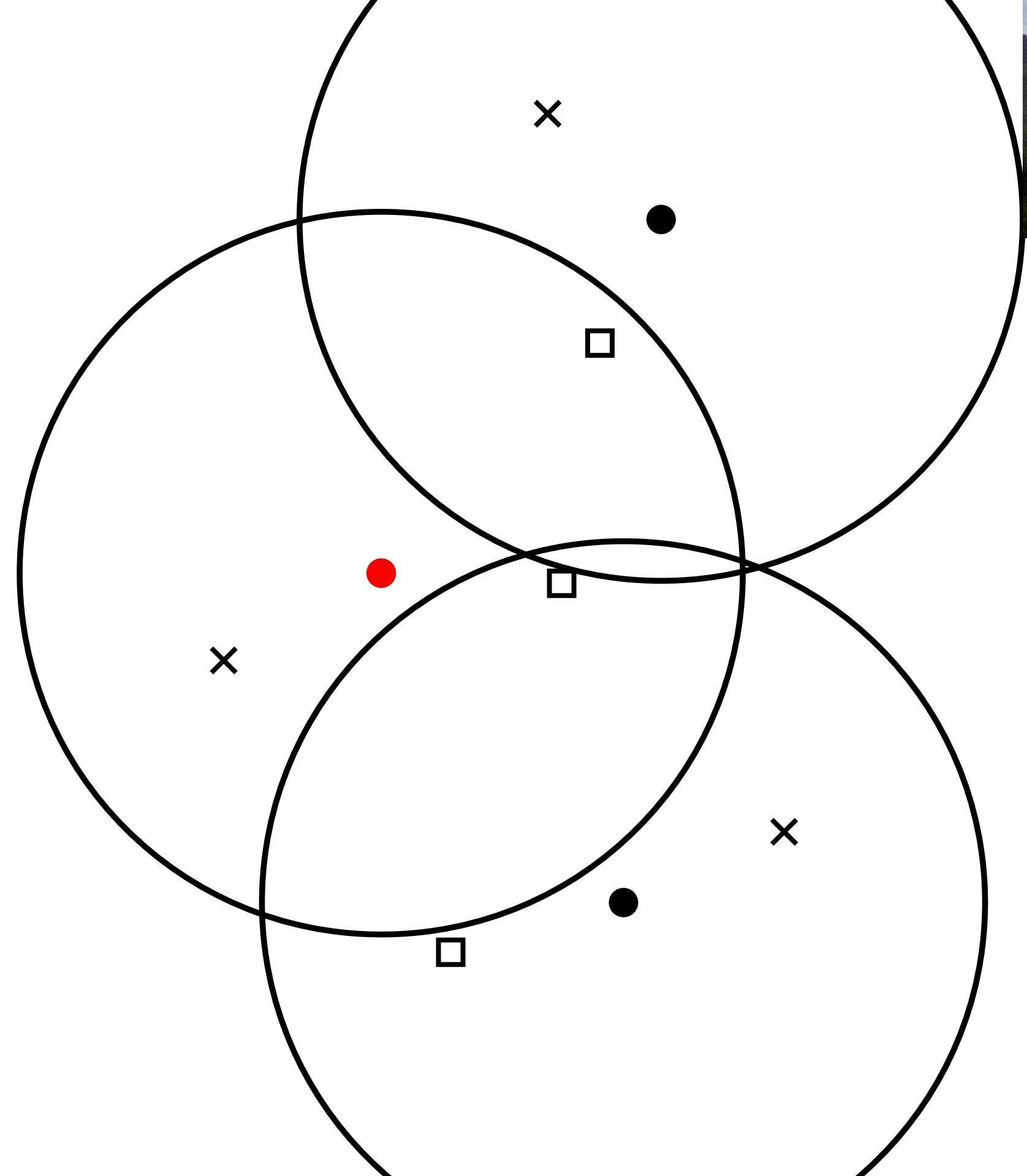
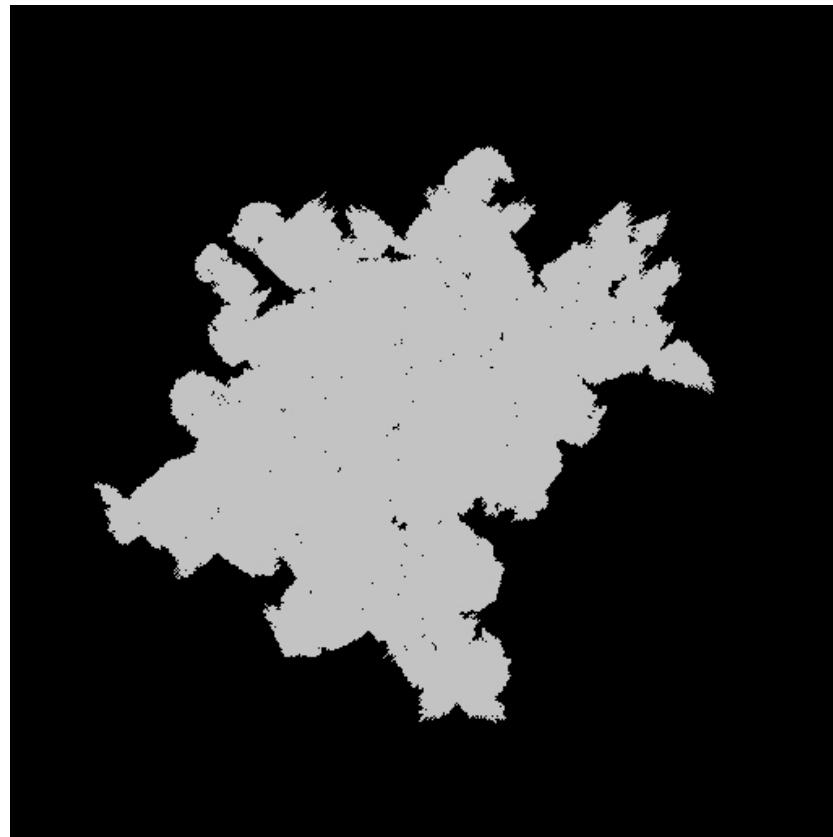


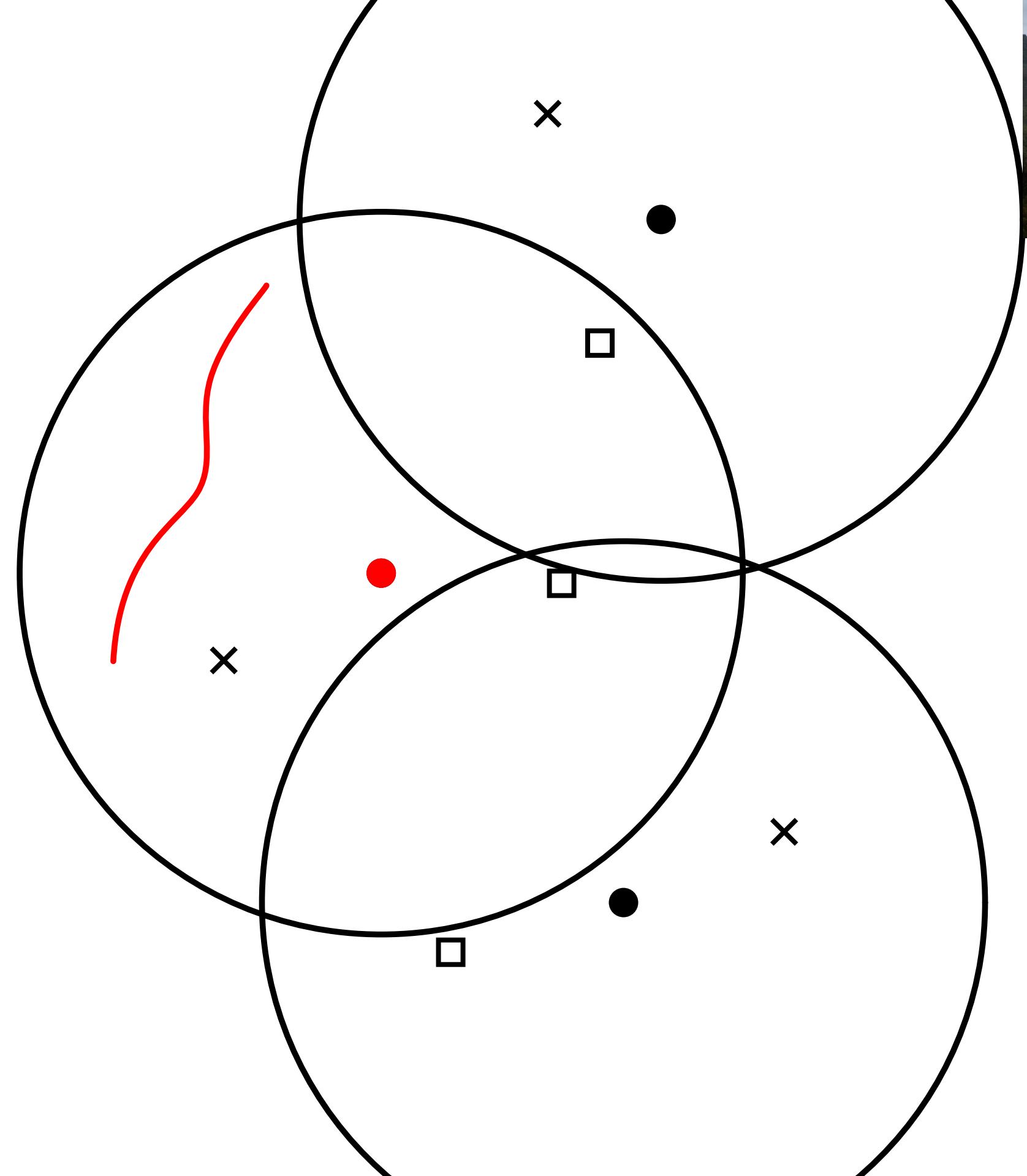
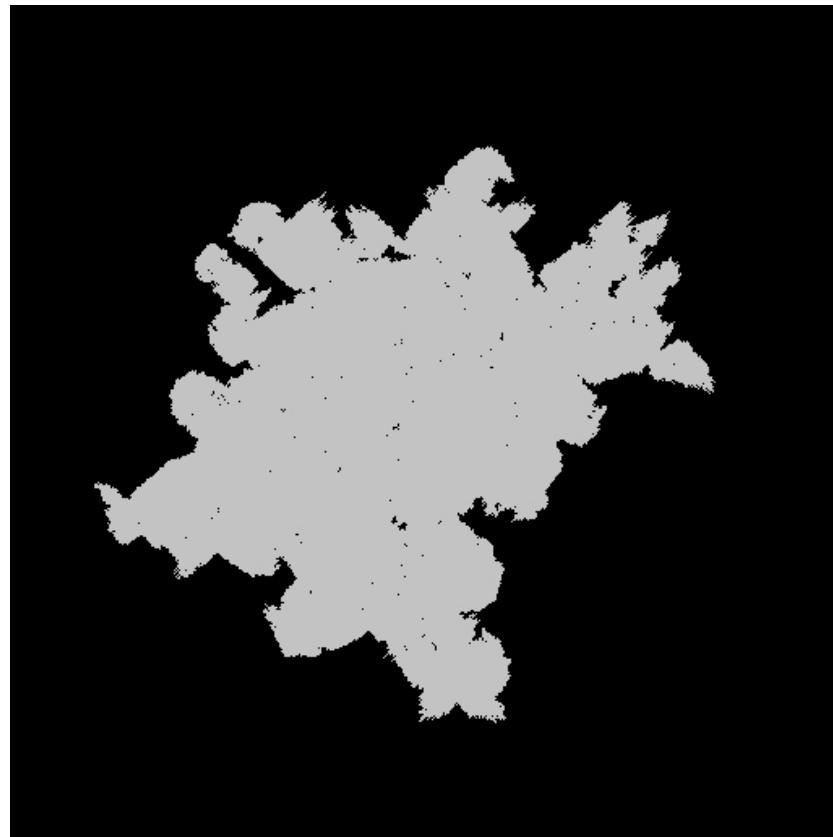


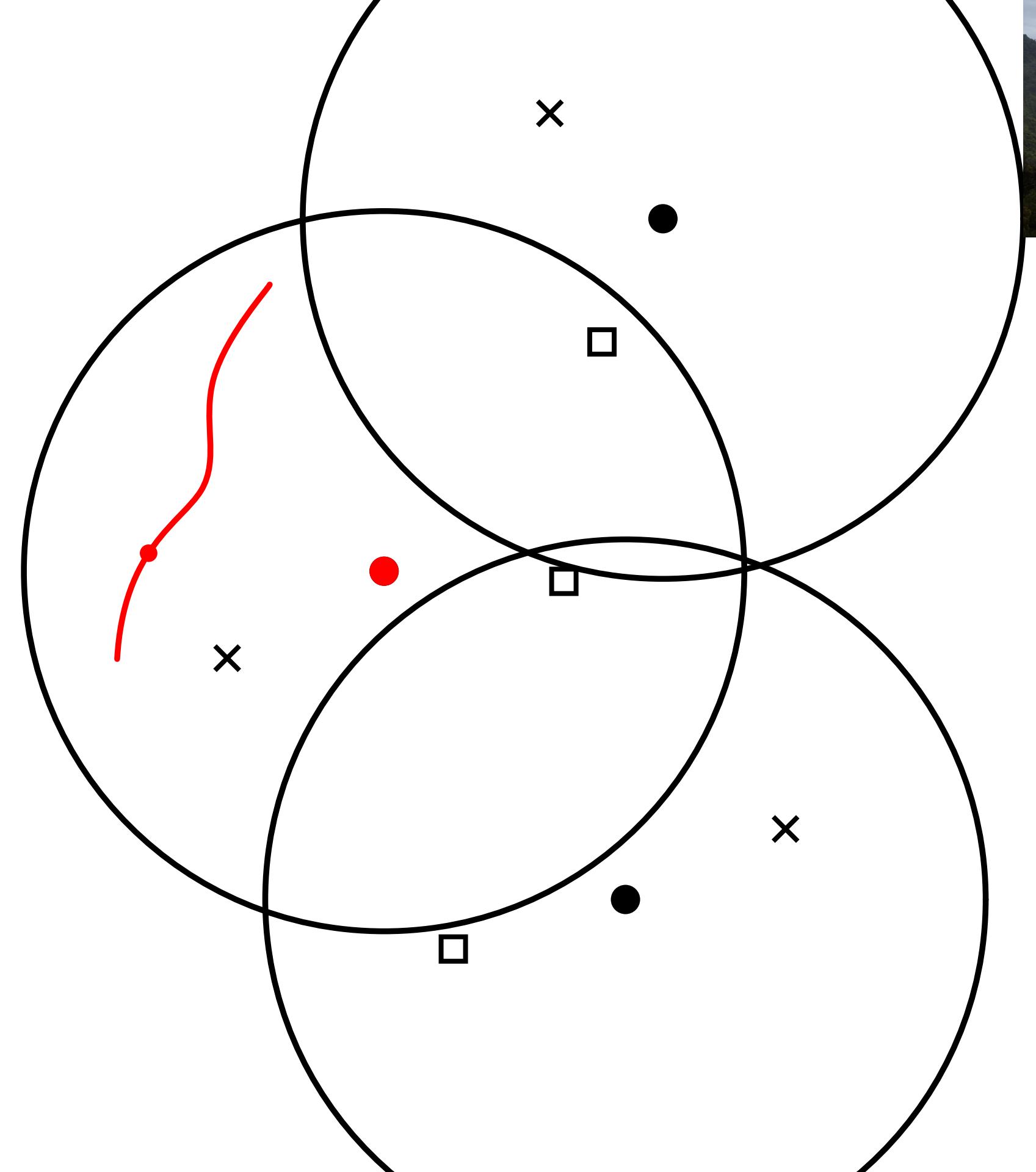
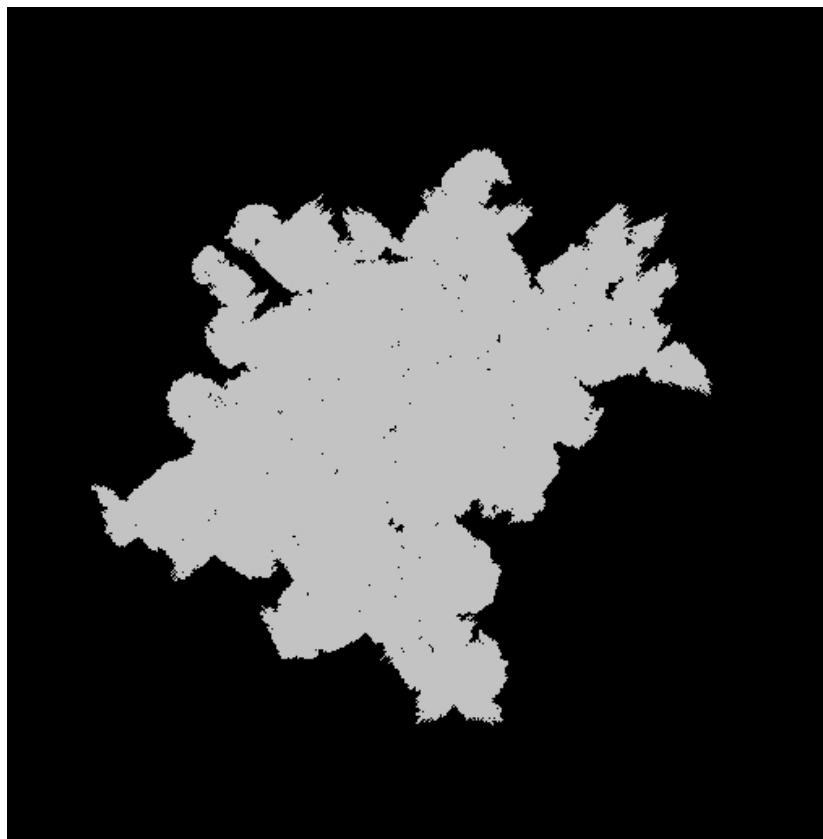


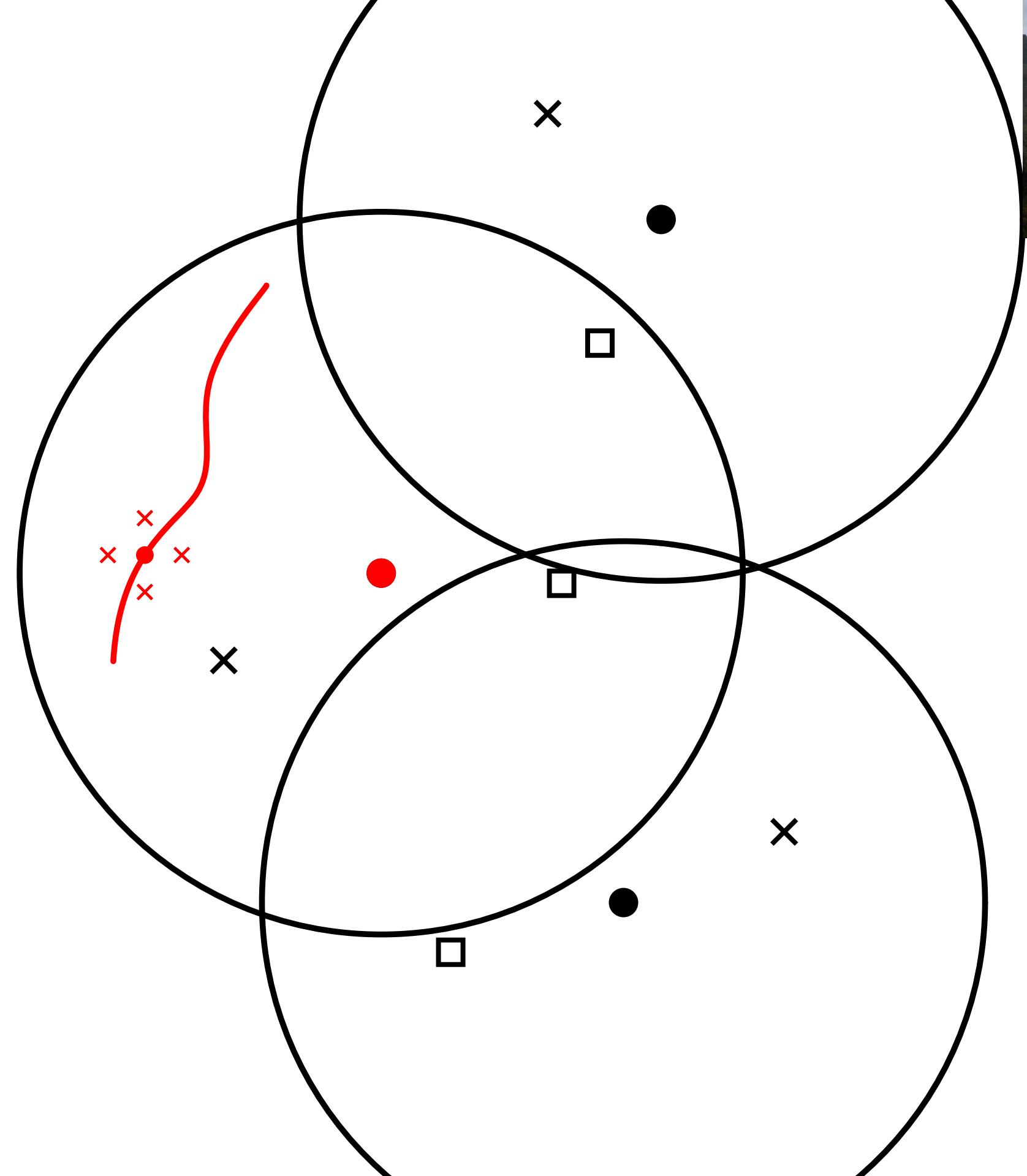
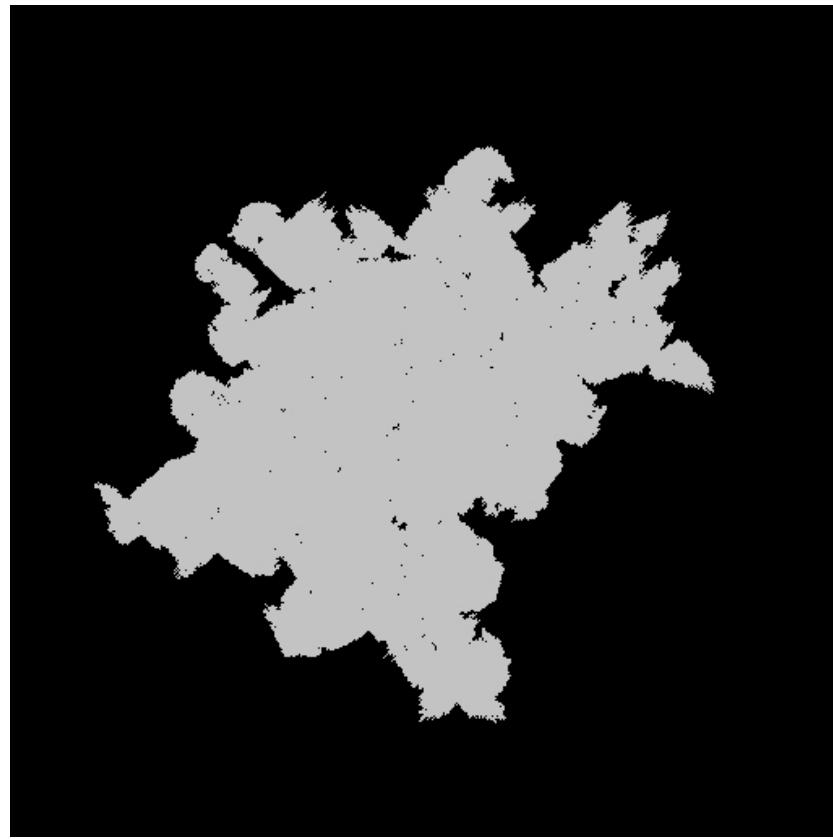


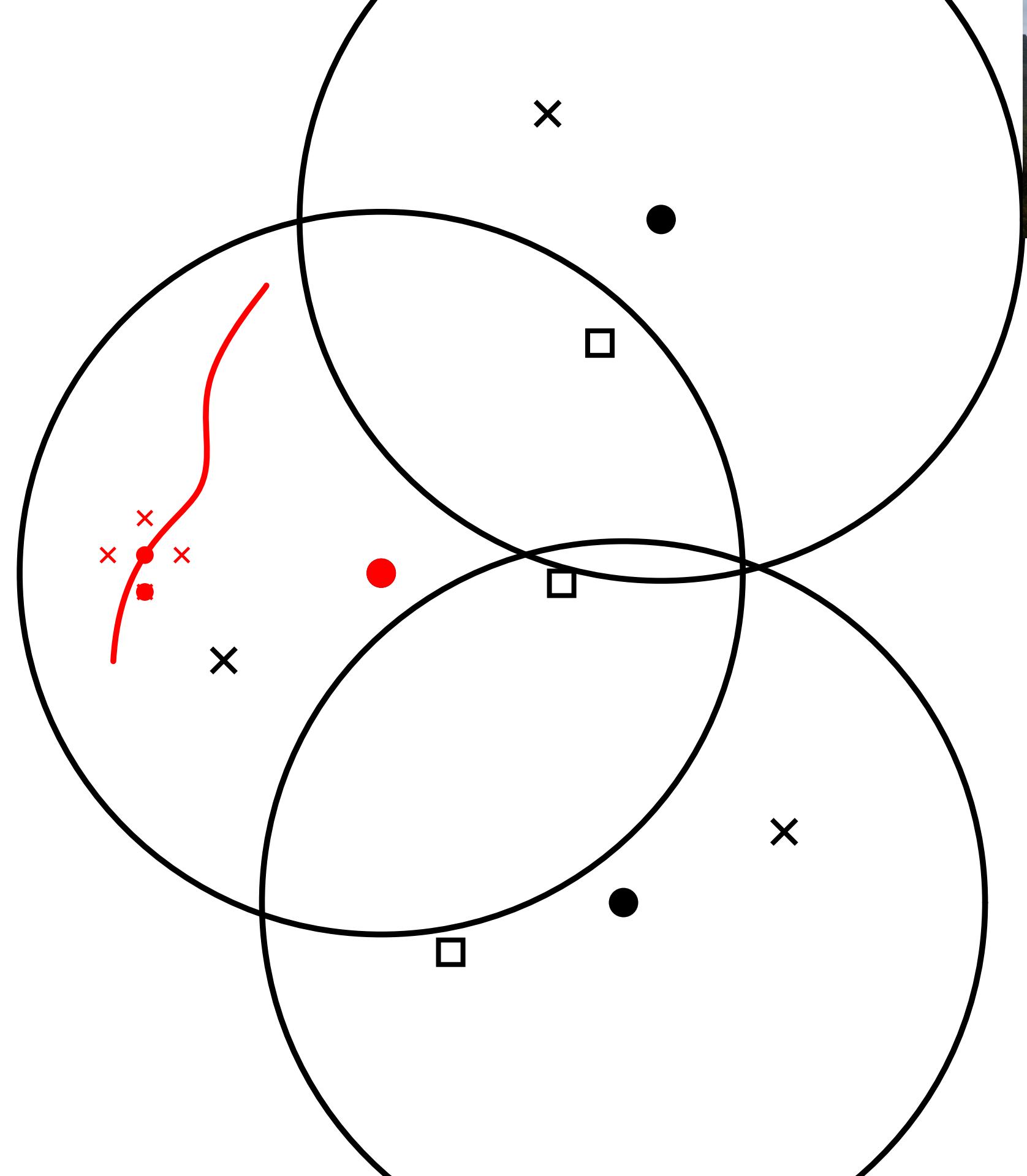
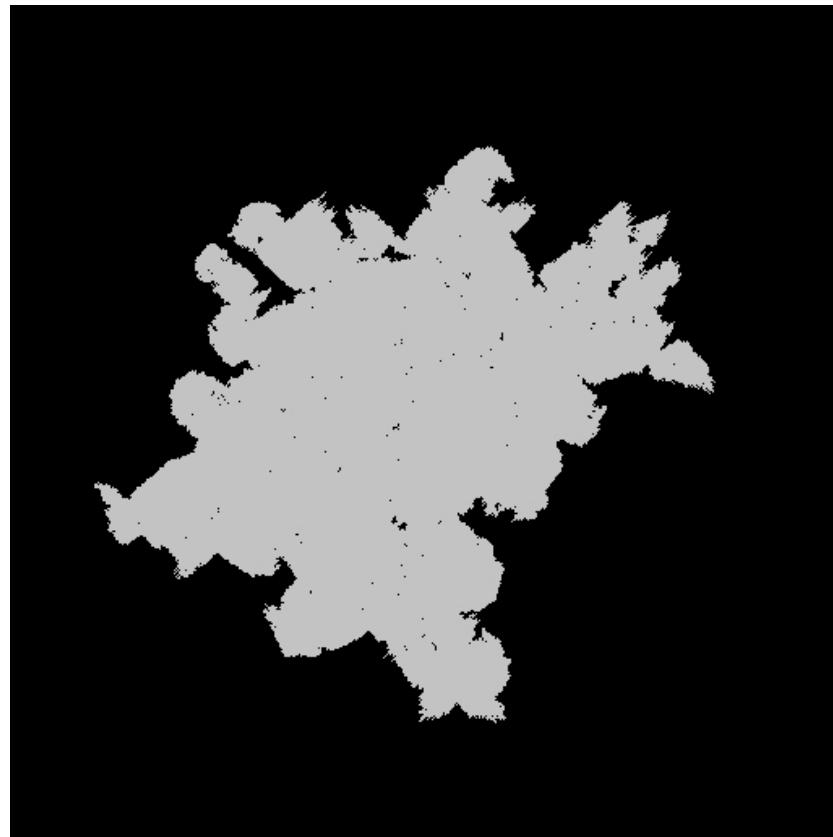


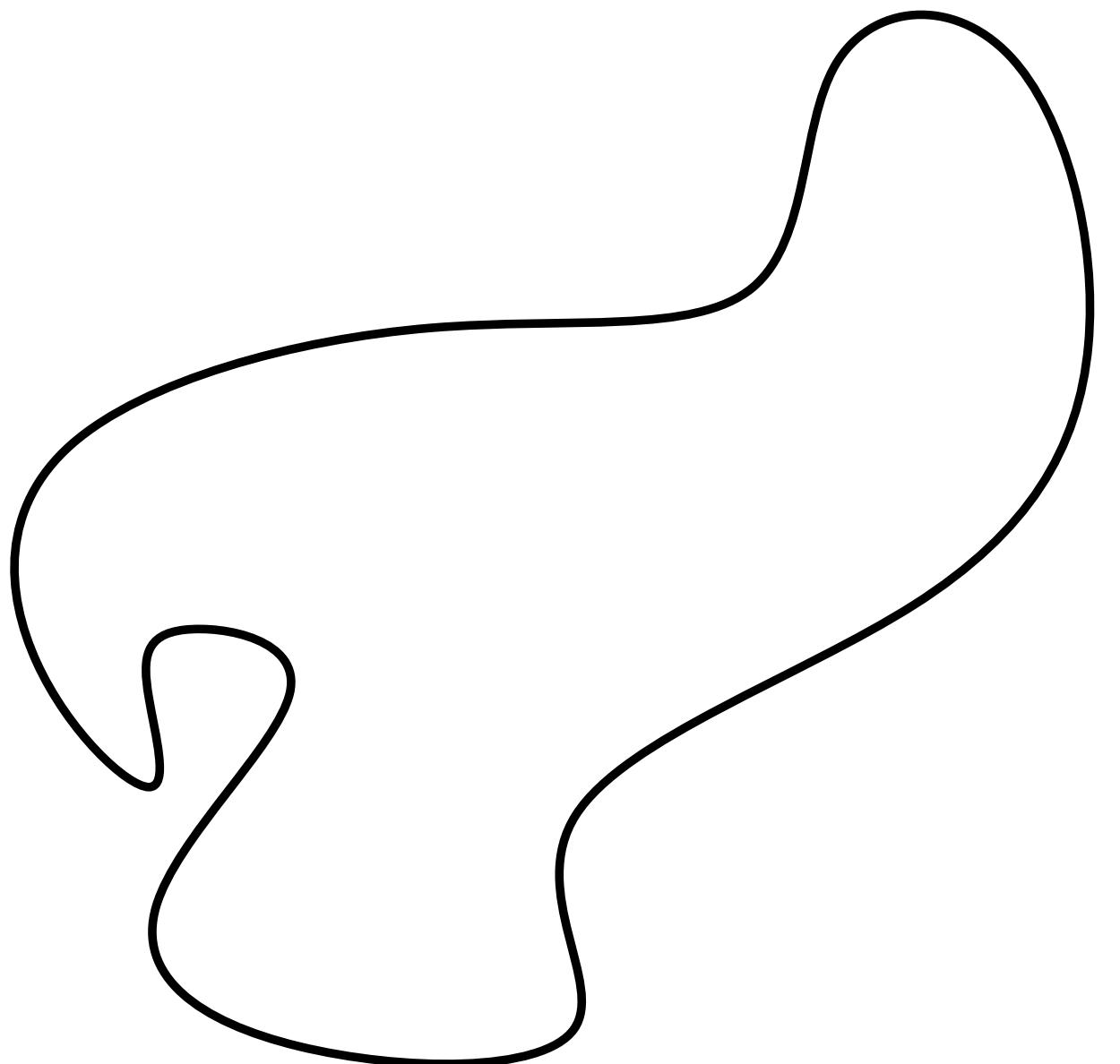


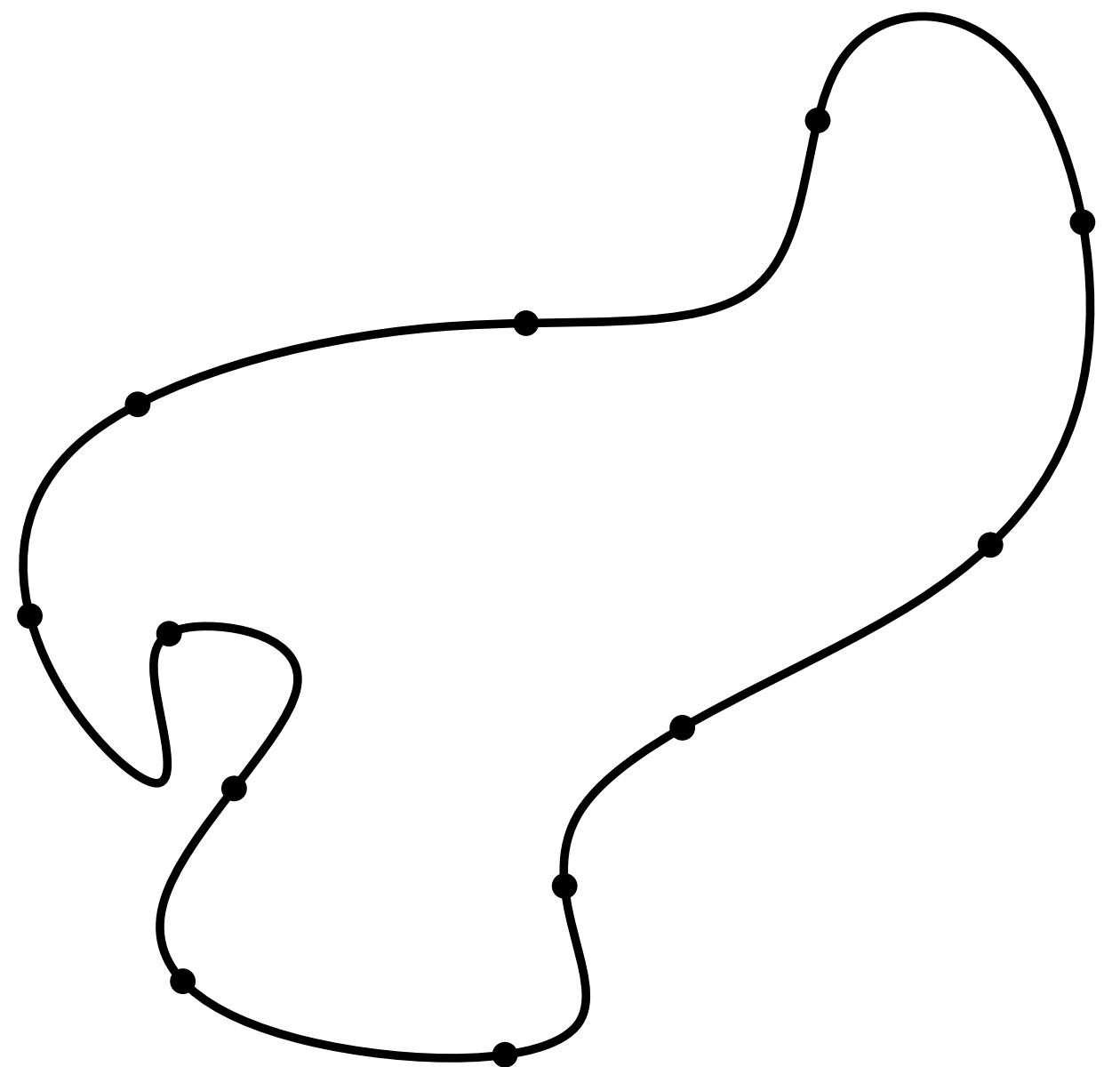


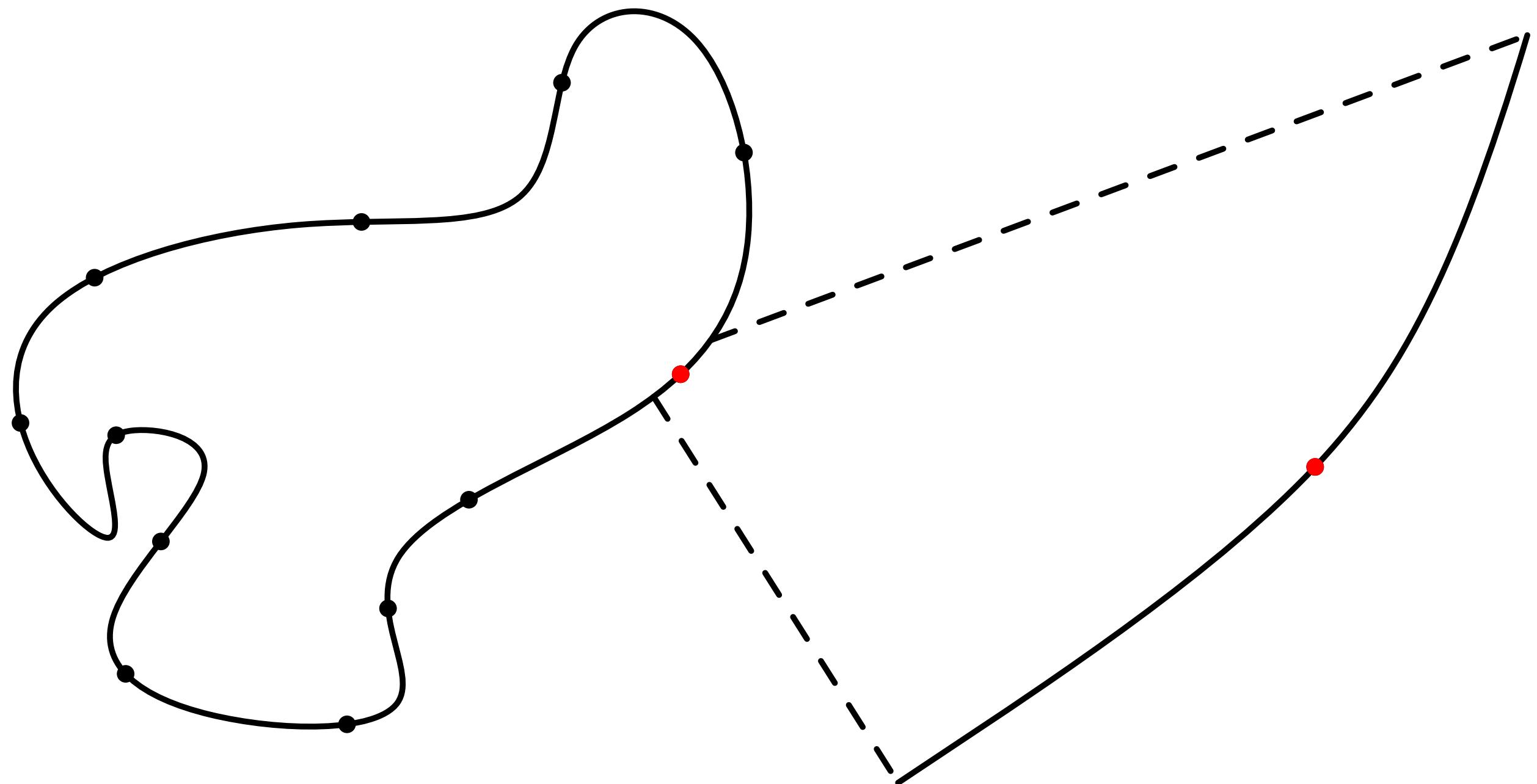


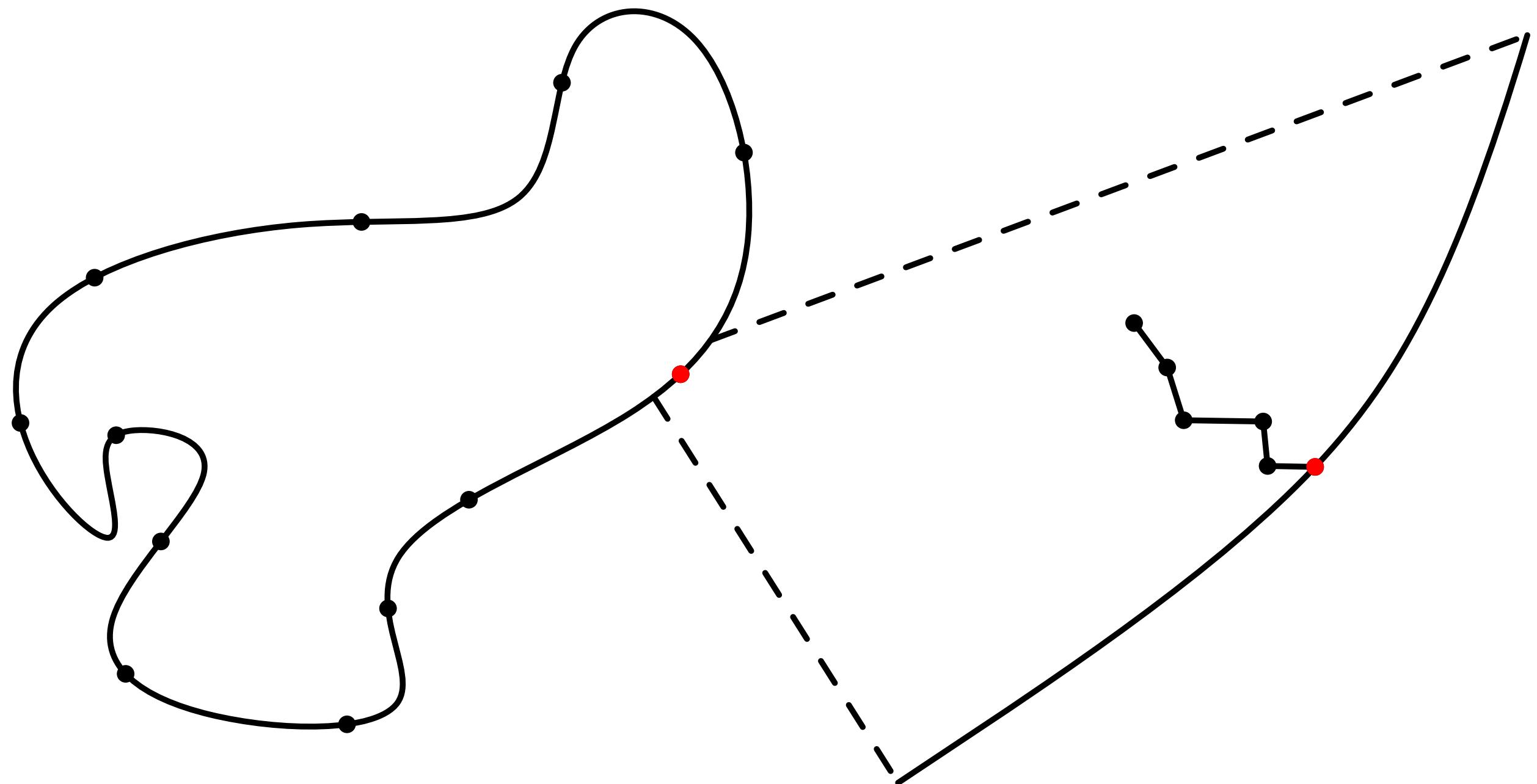


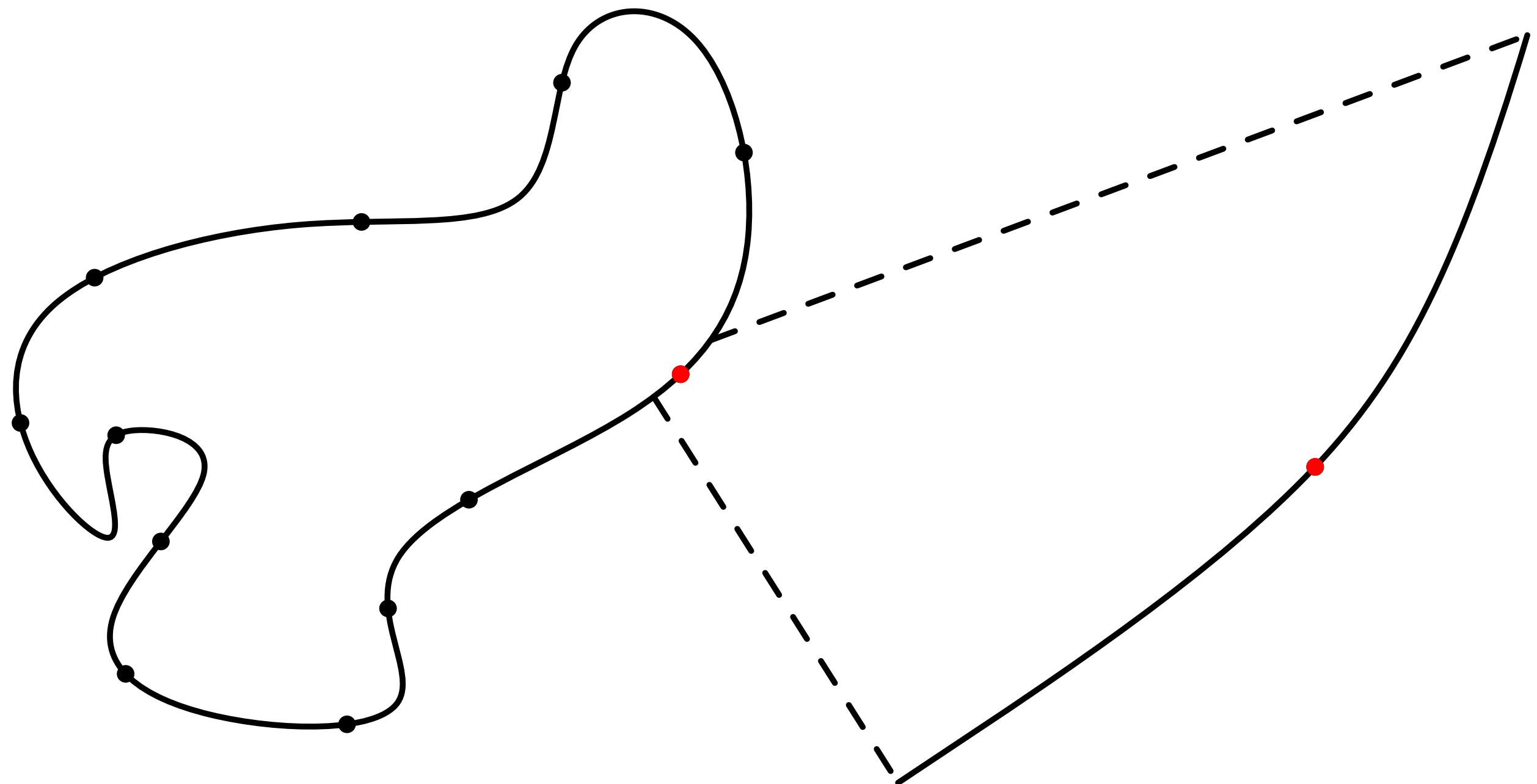


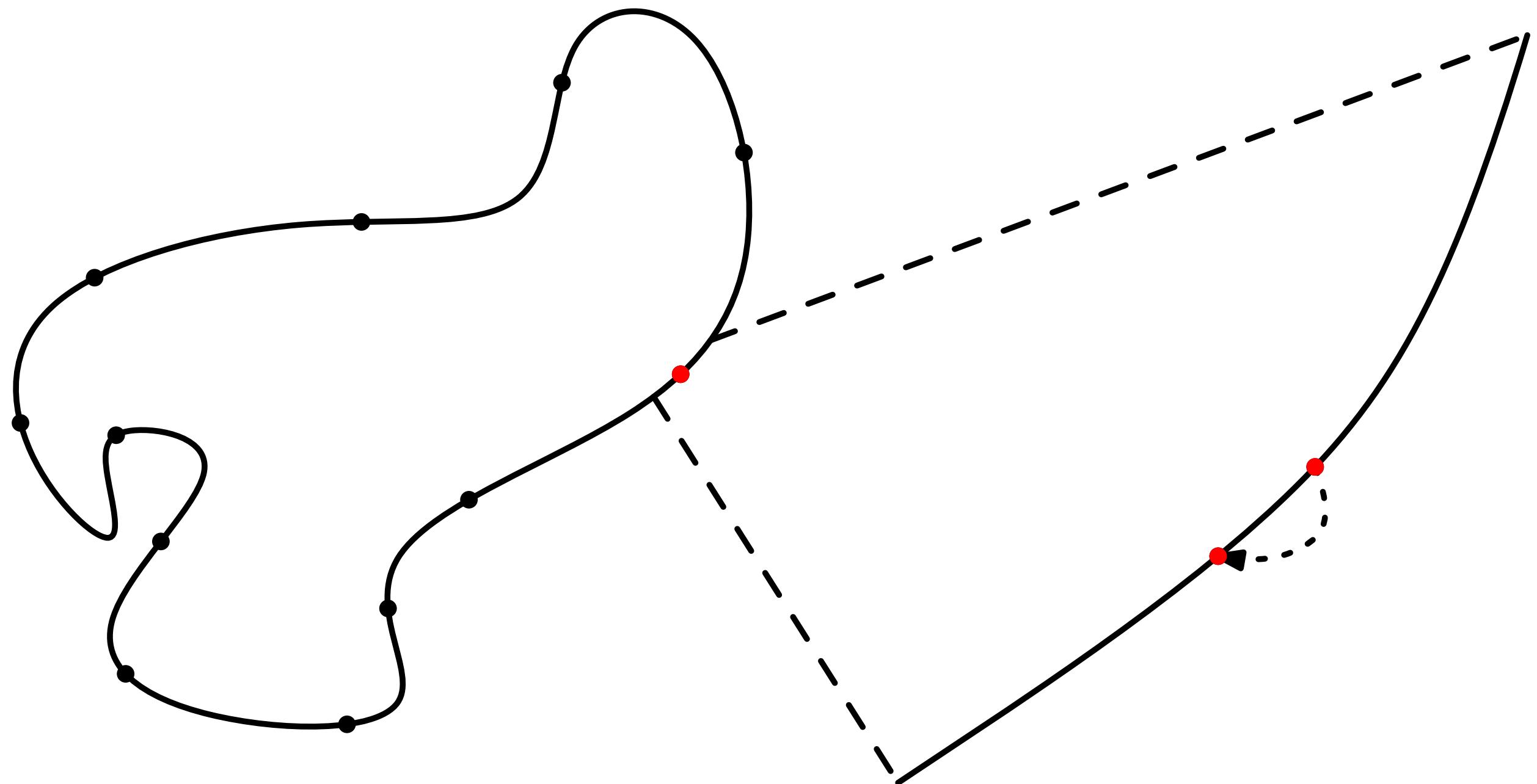






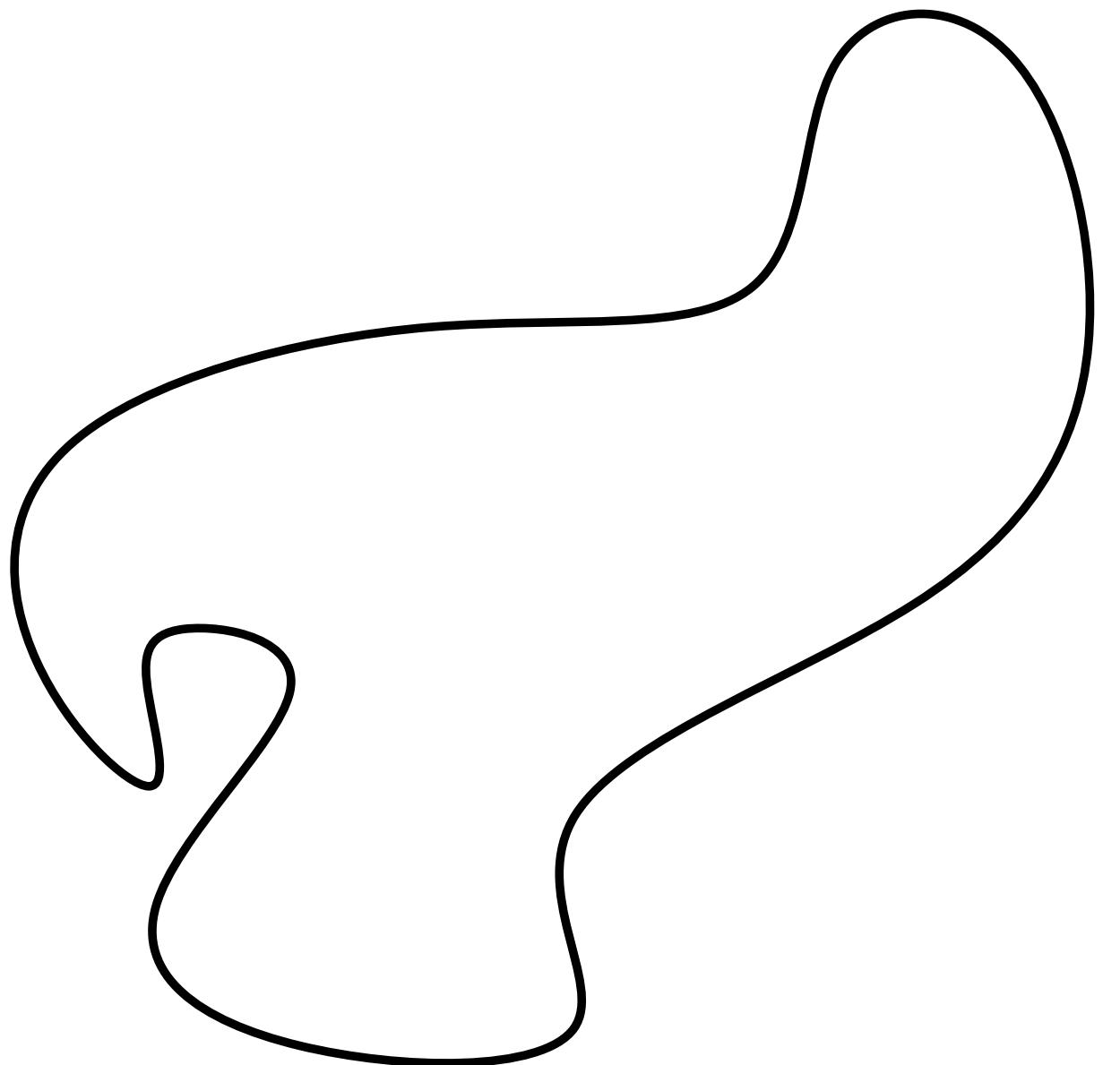


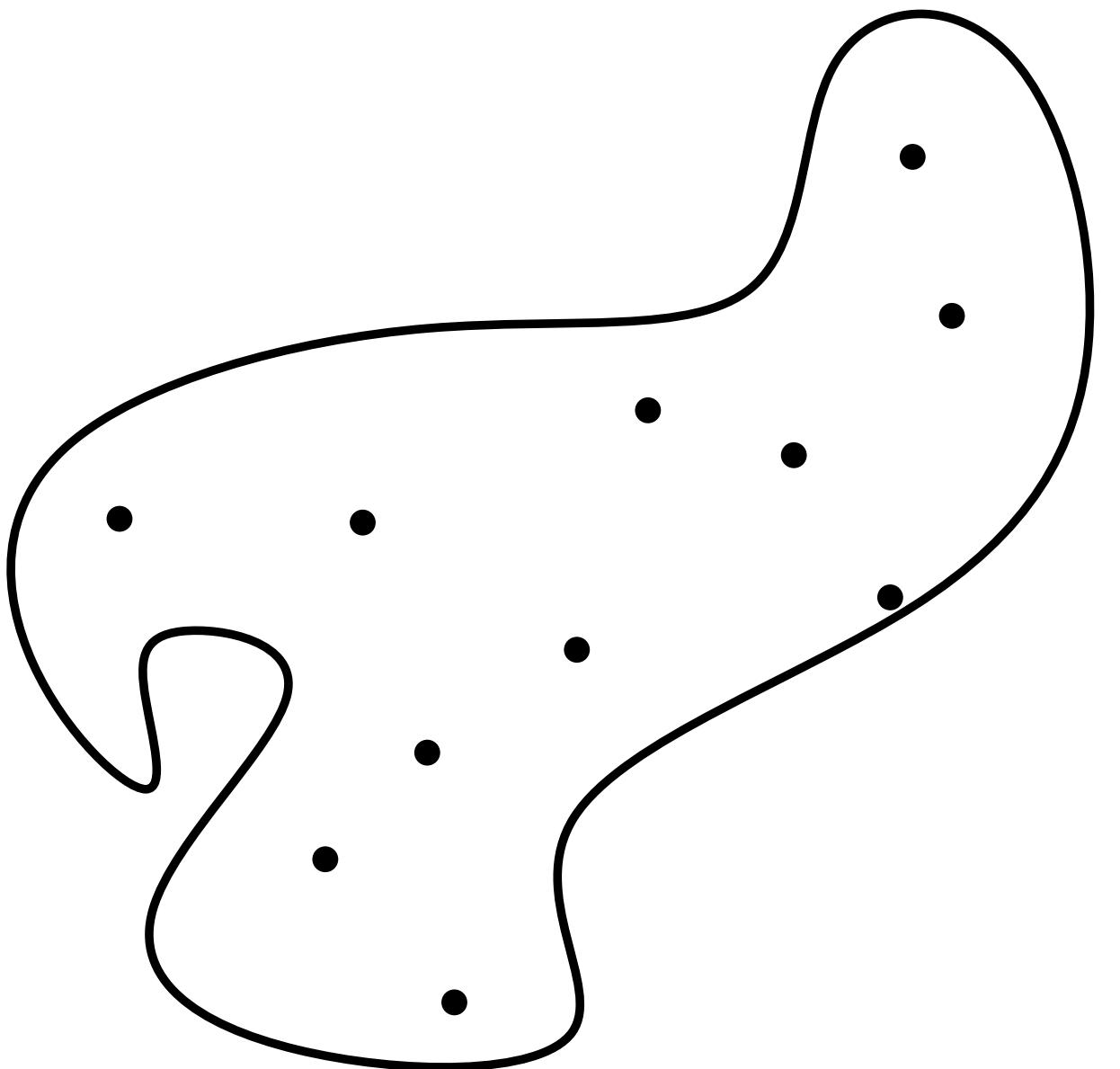


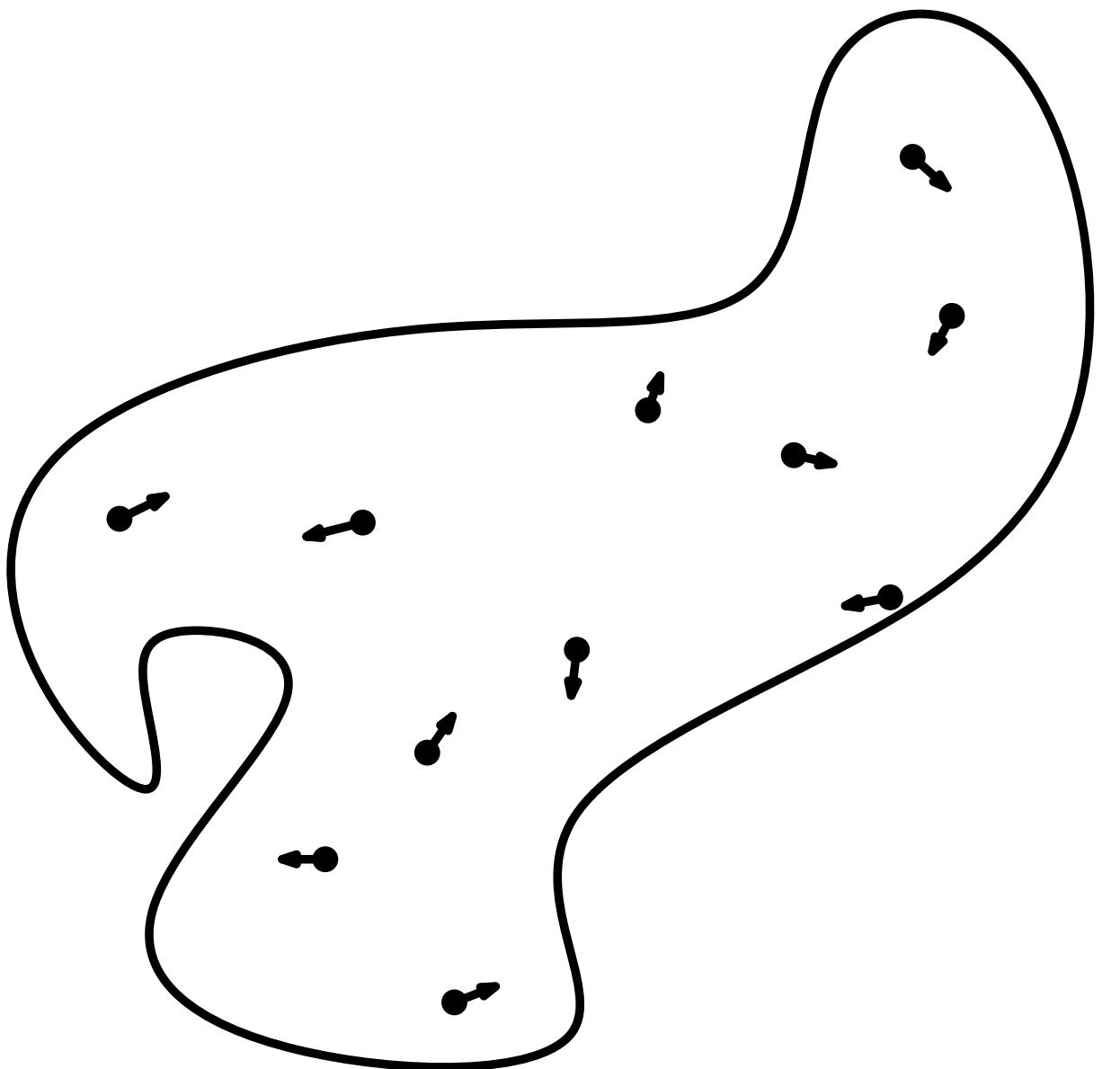


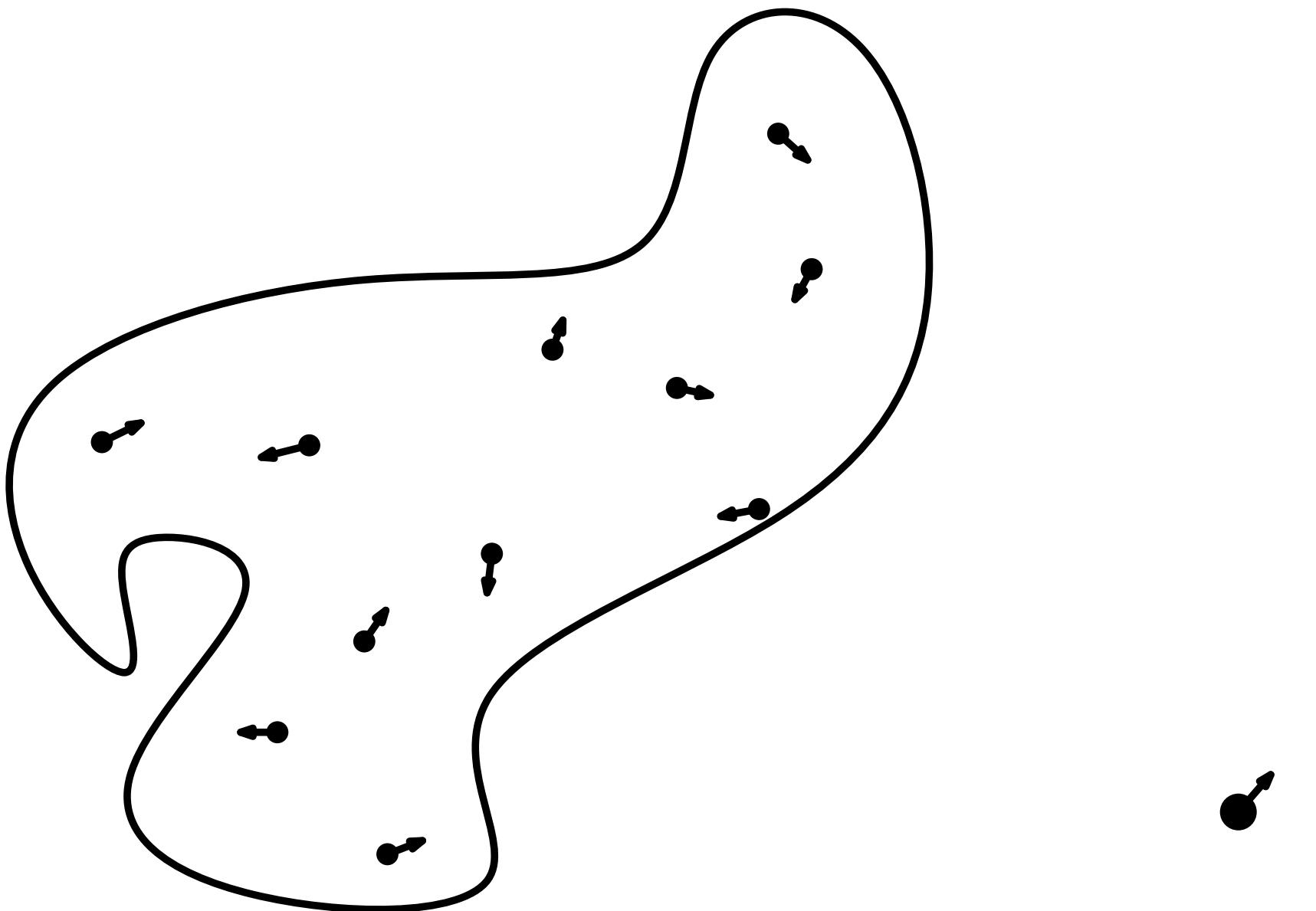
Assignment 3

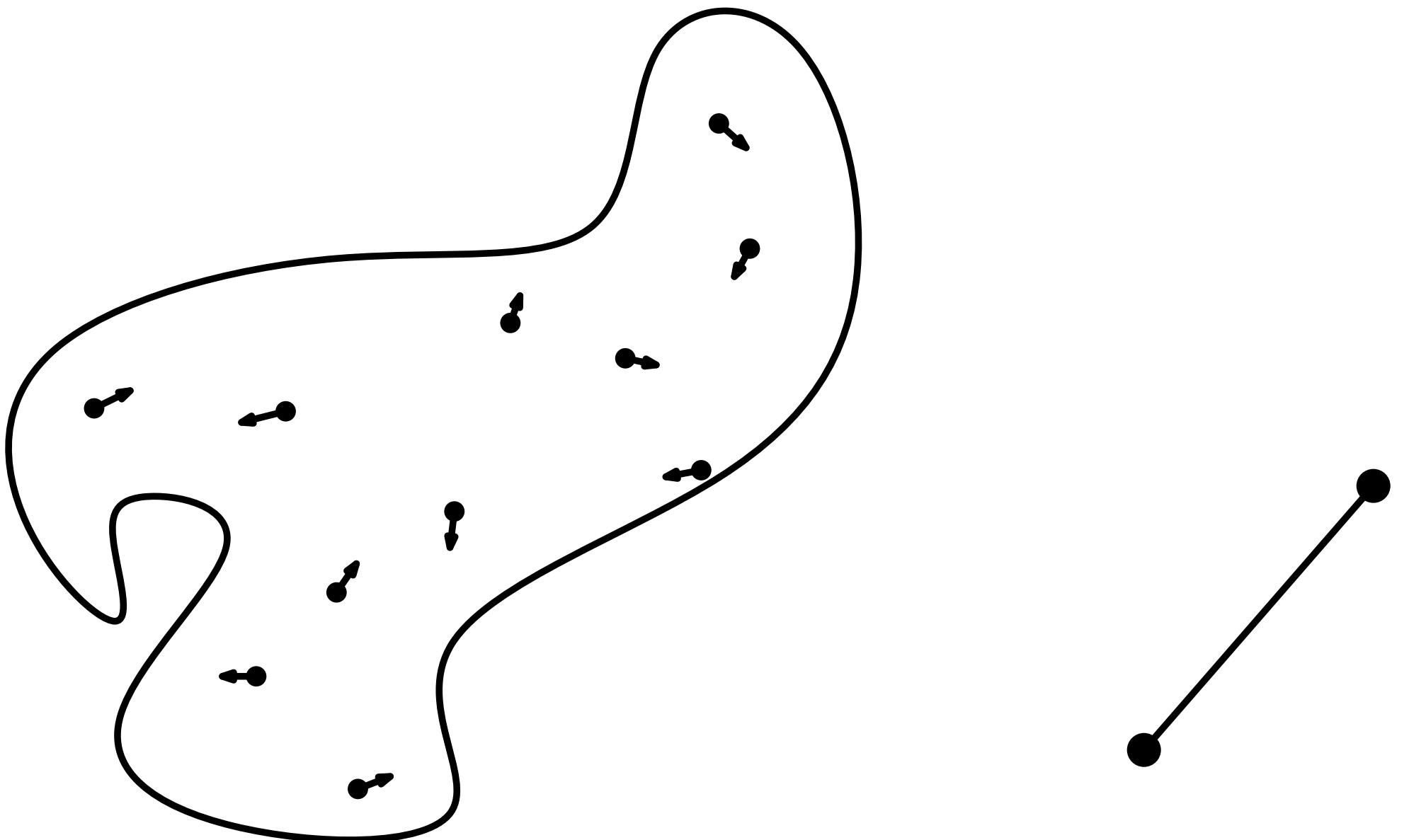
The beach agents as described places beaches in random locations.
How could we modify the agents such that the beaches are placed in more realistic locations?

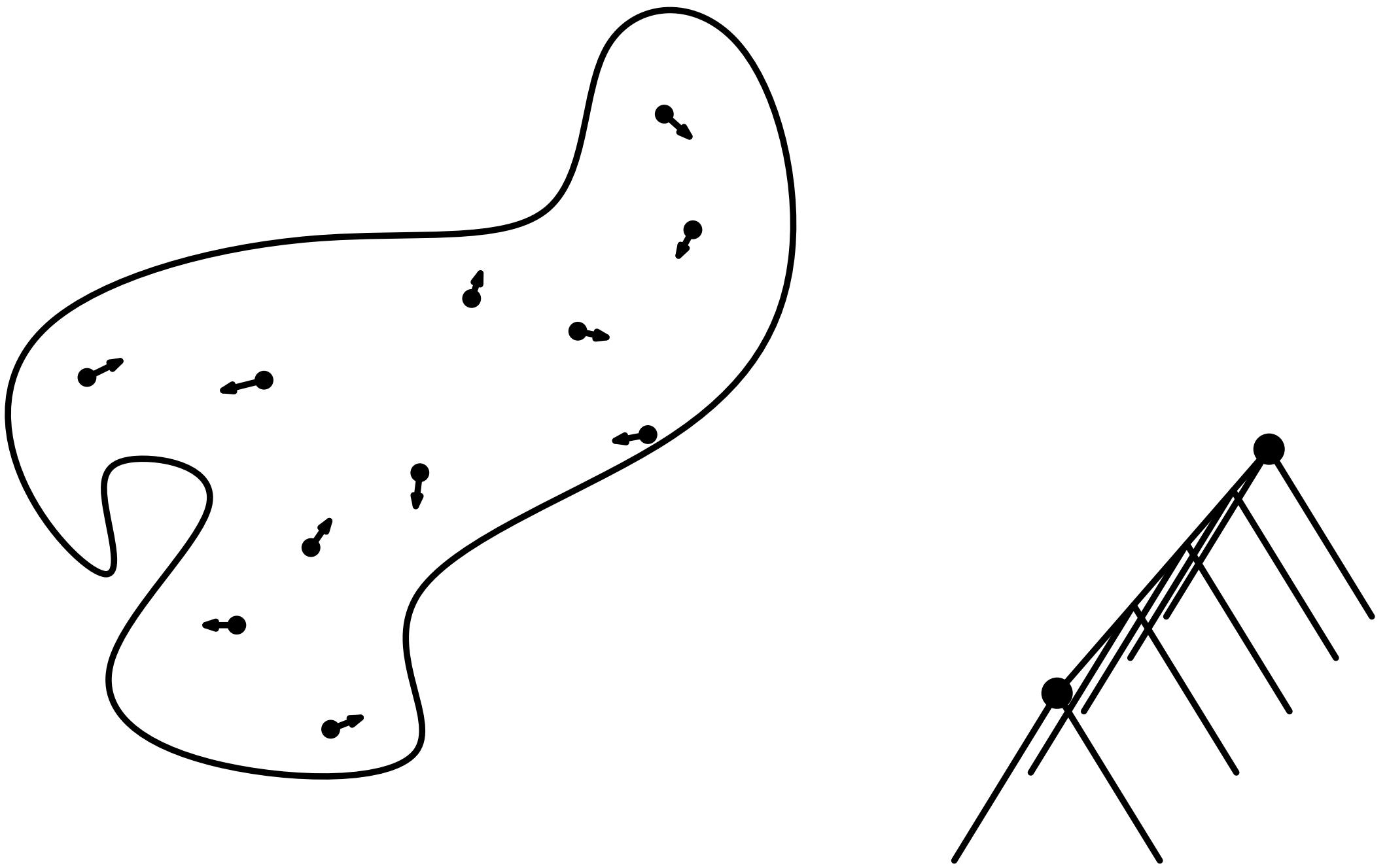


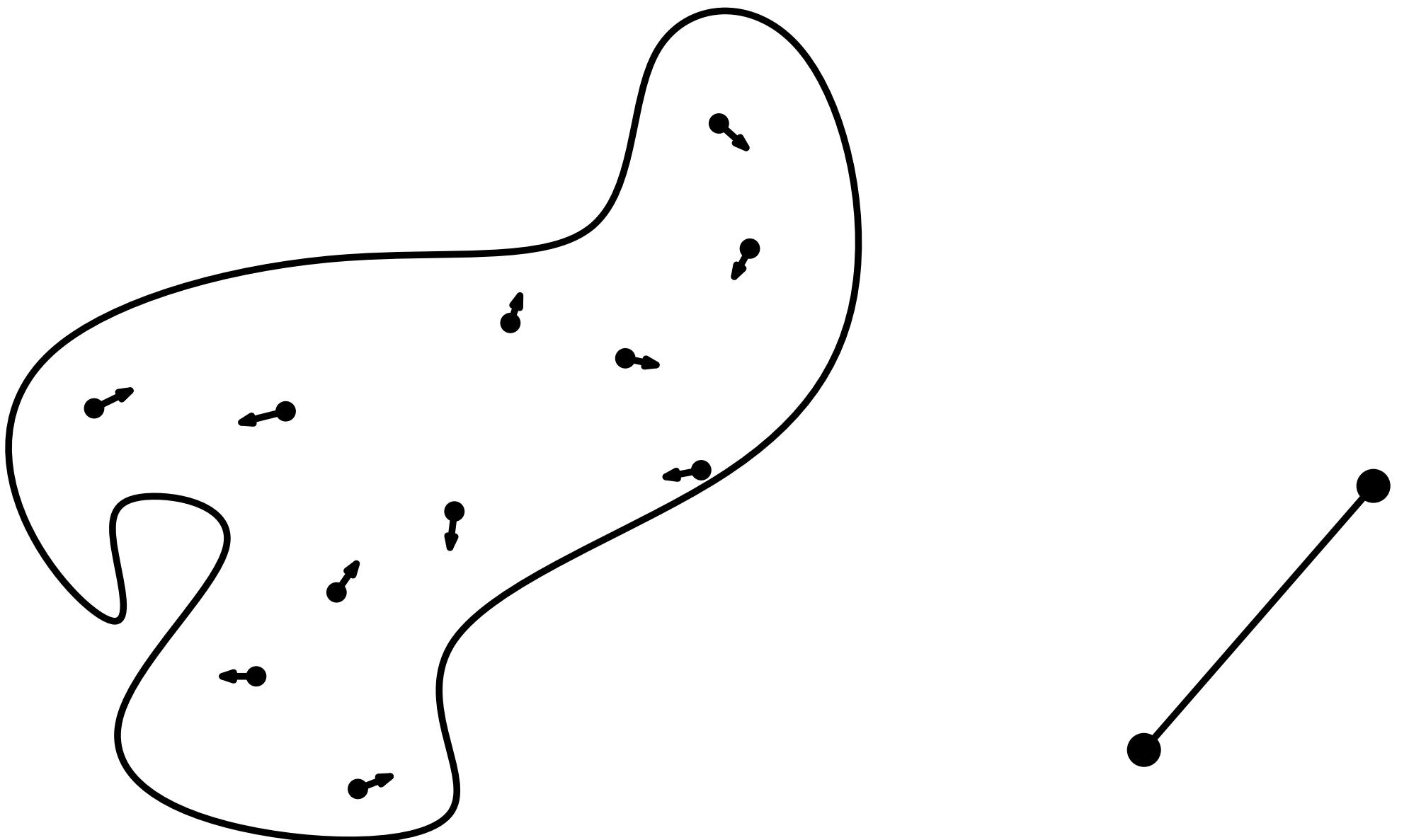


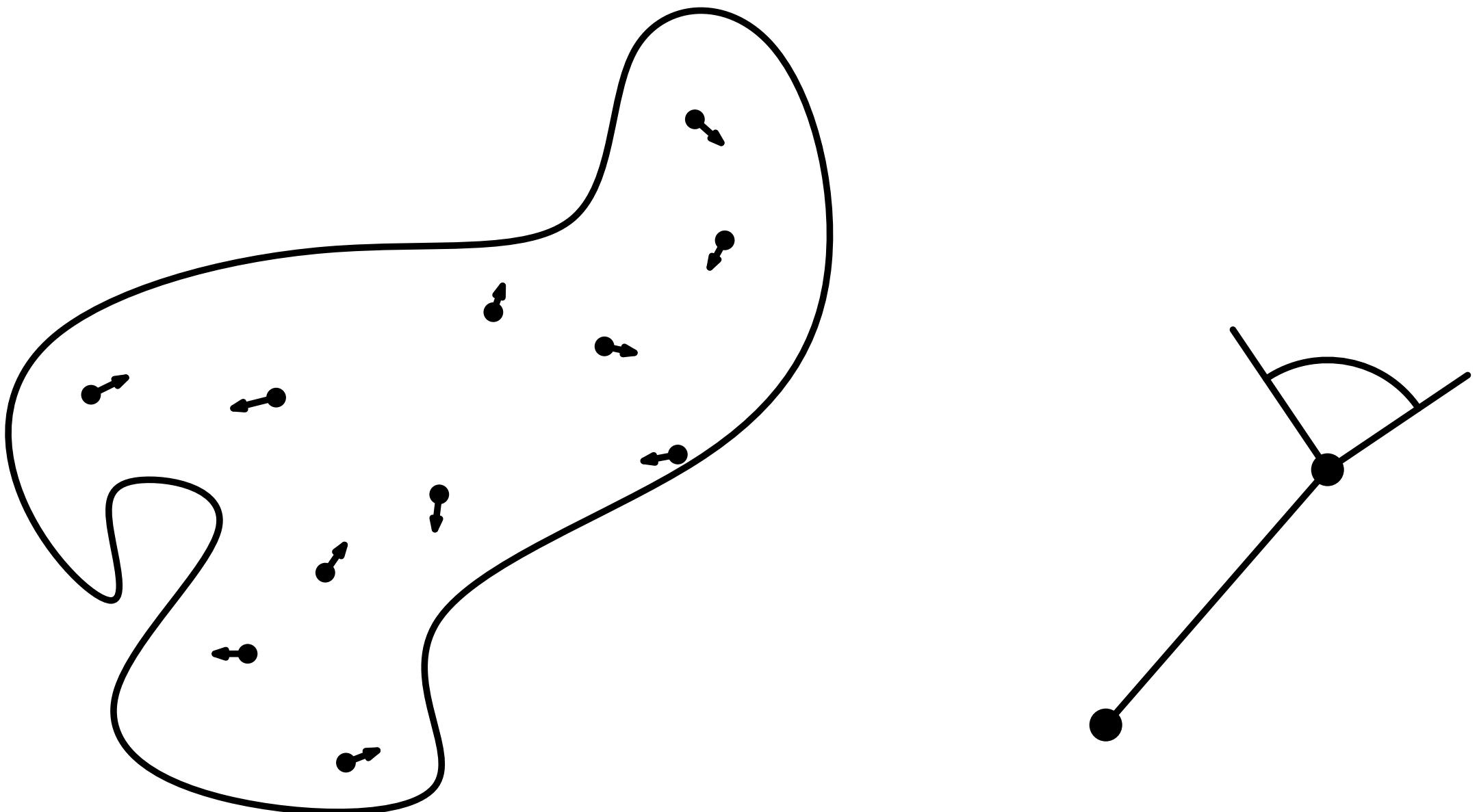


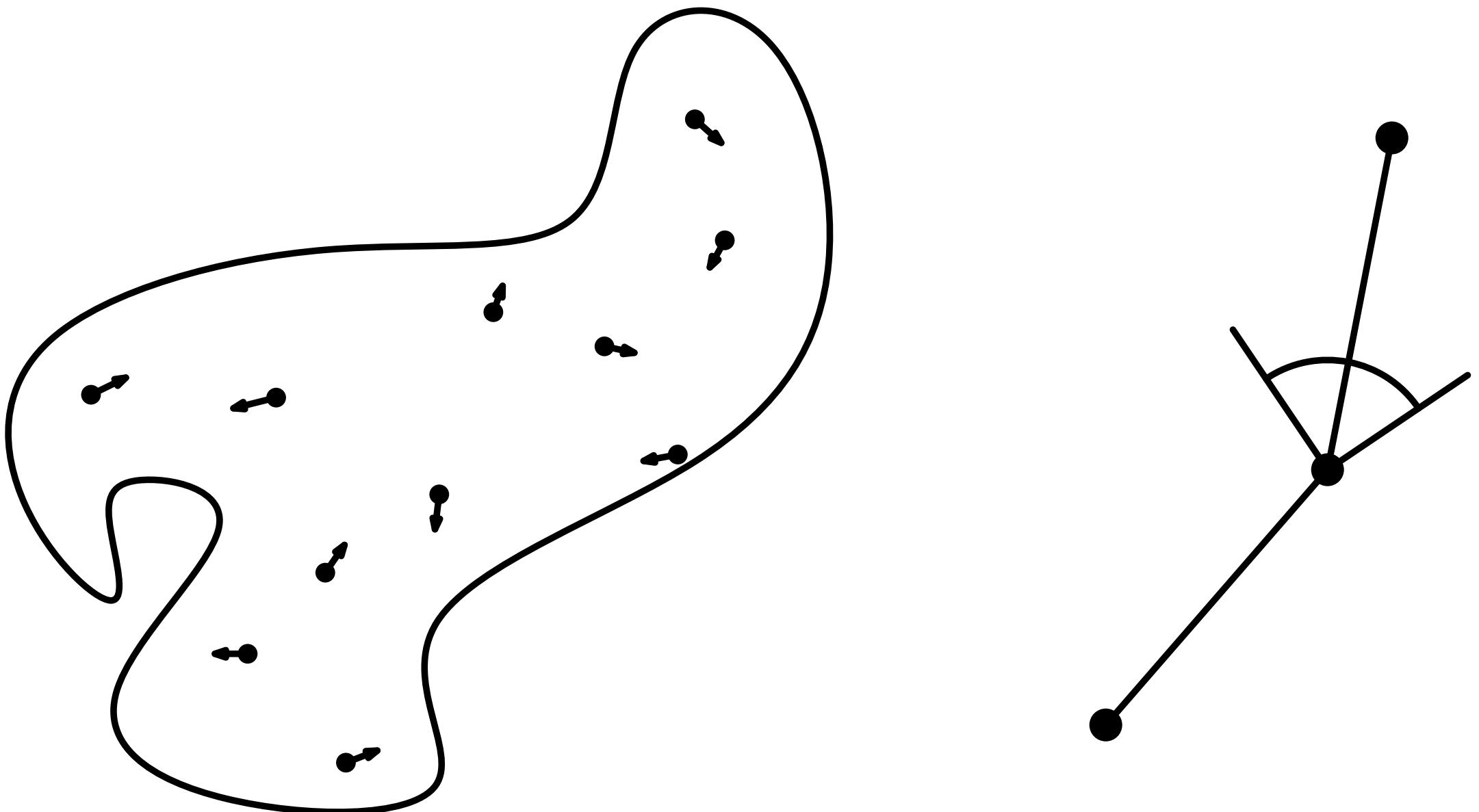


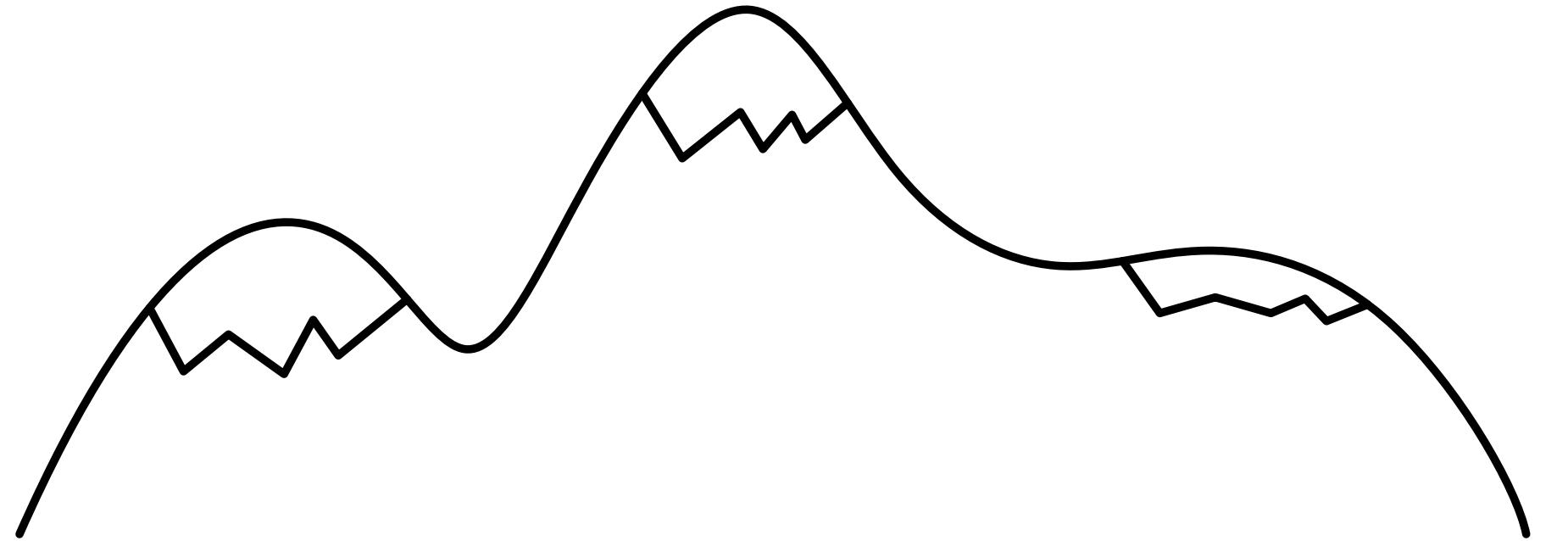


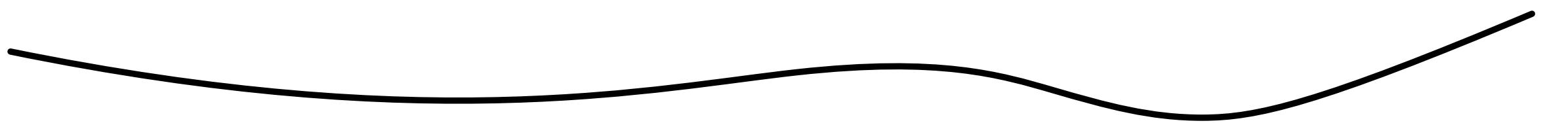
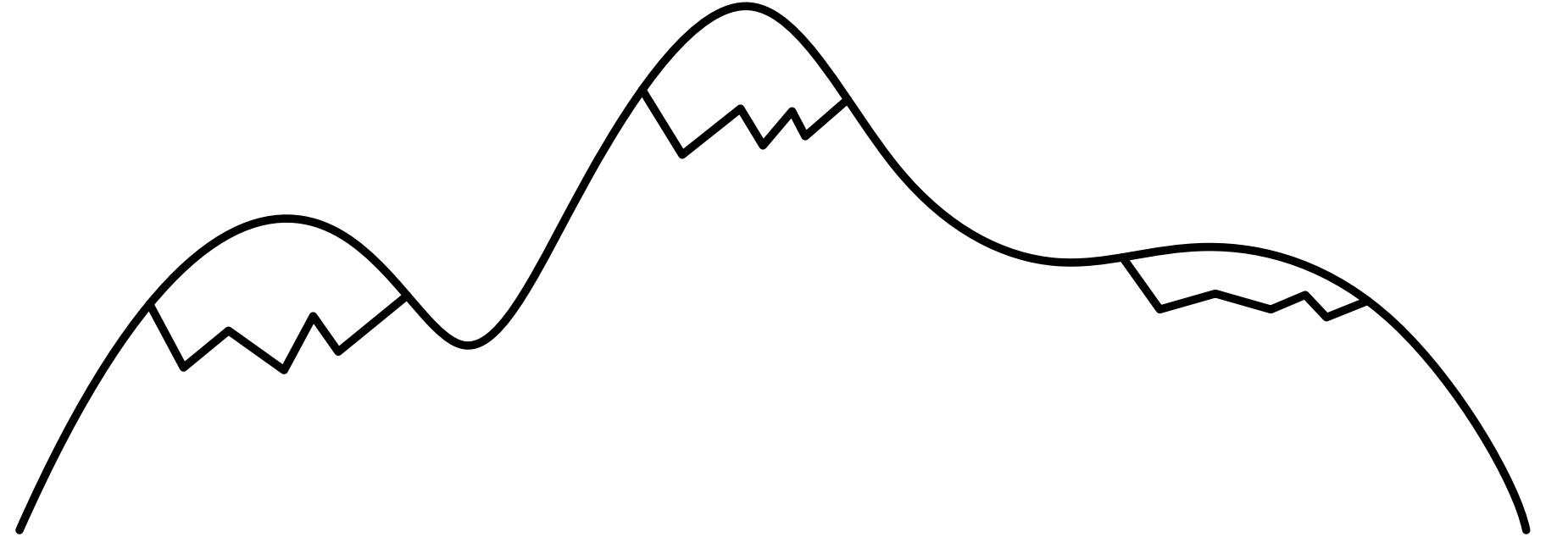


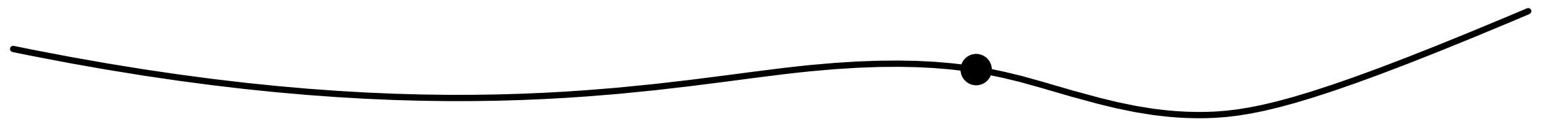
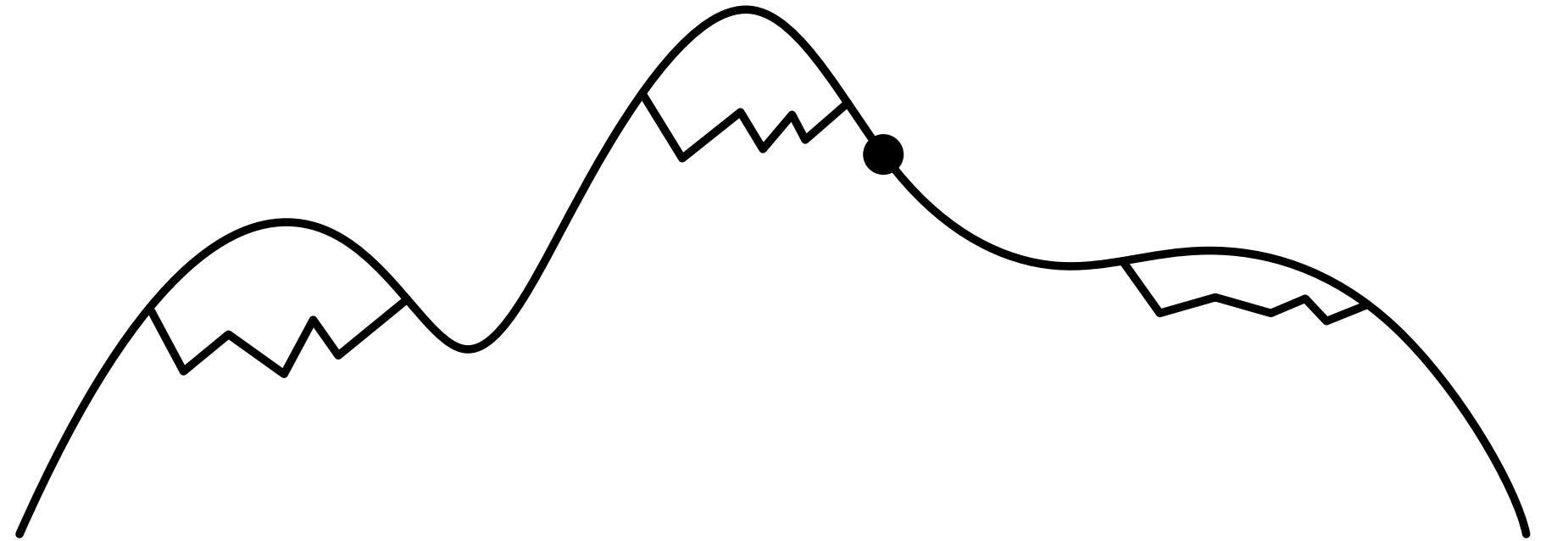


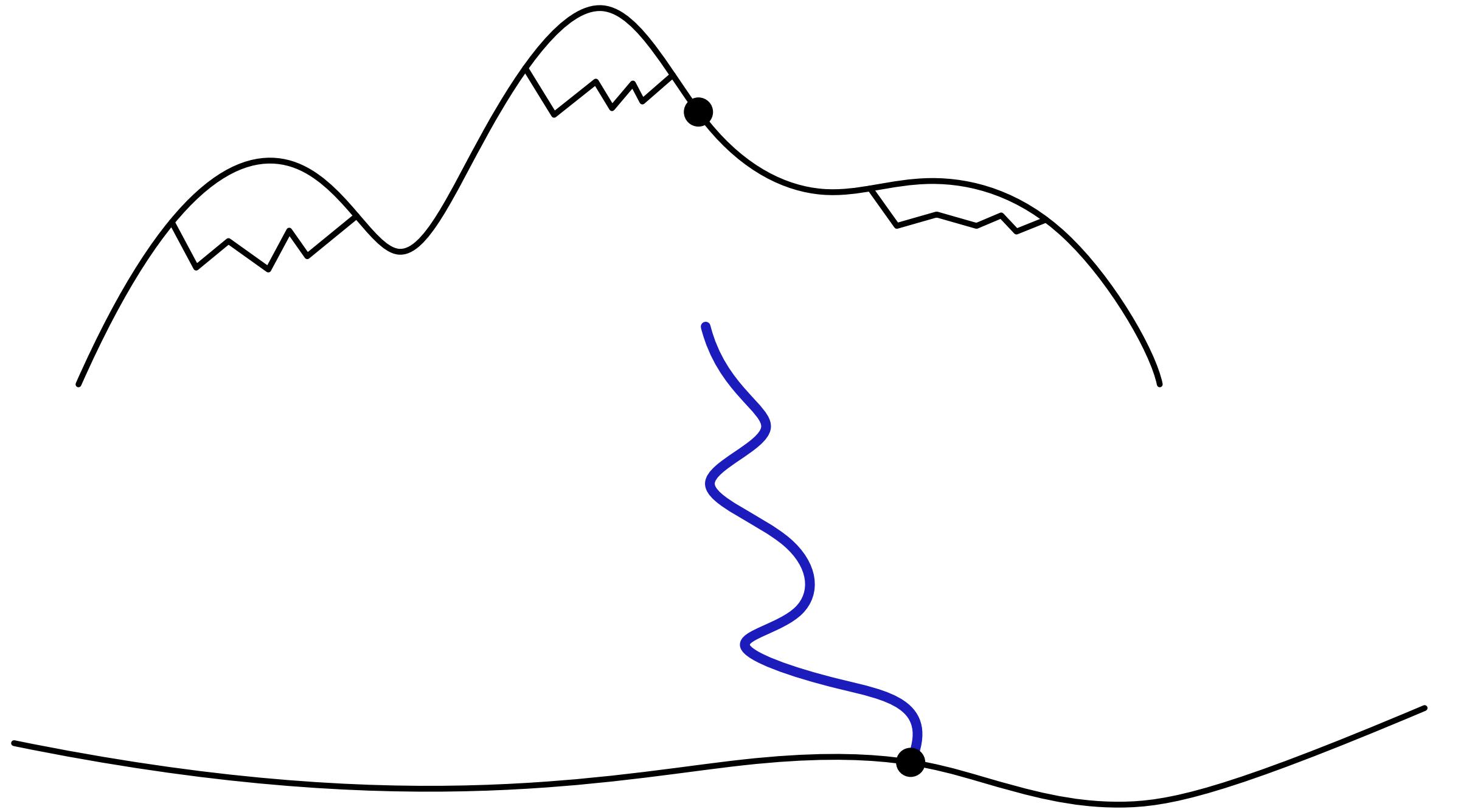


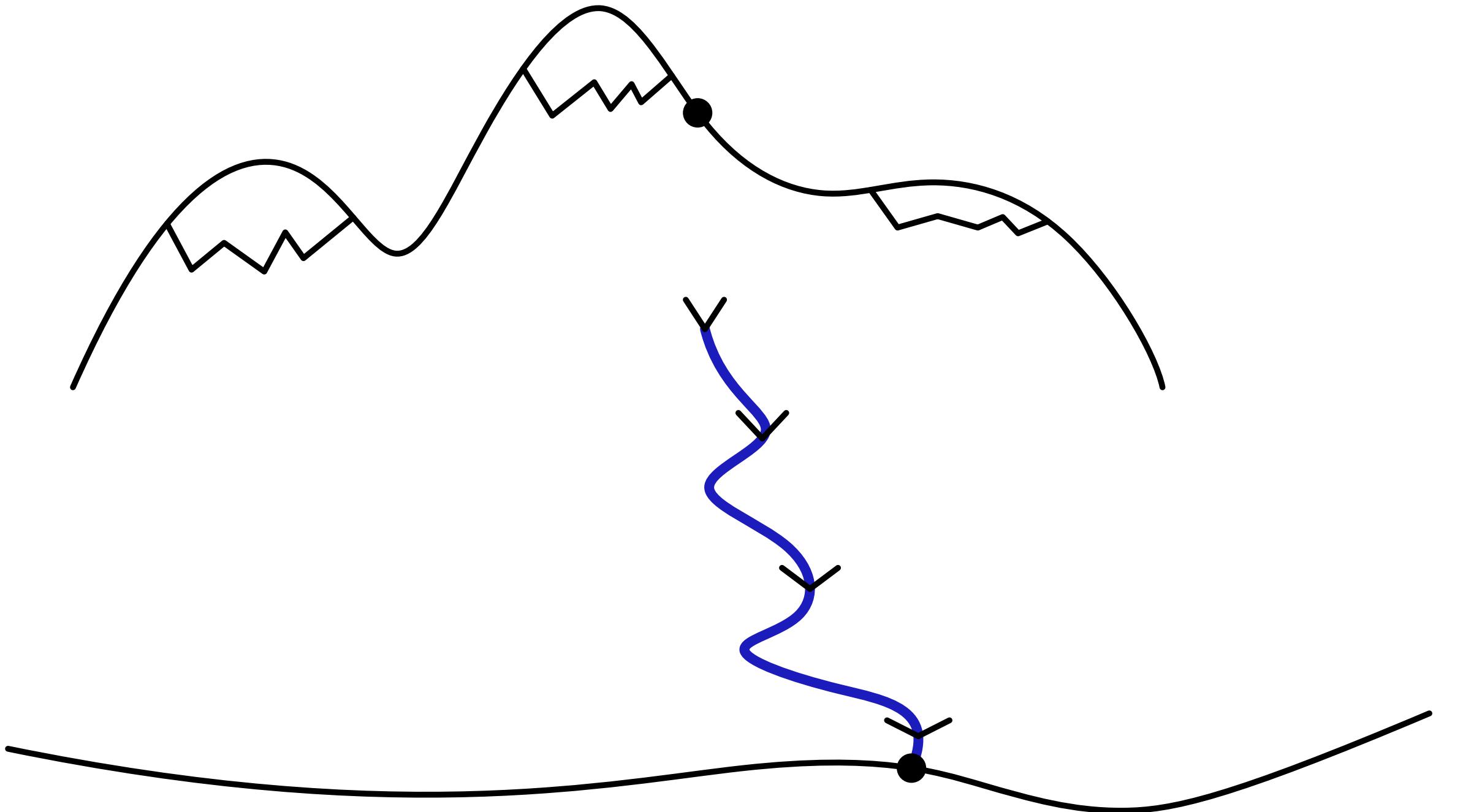


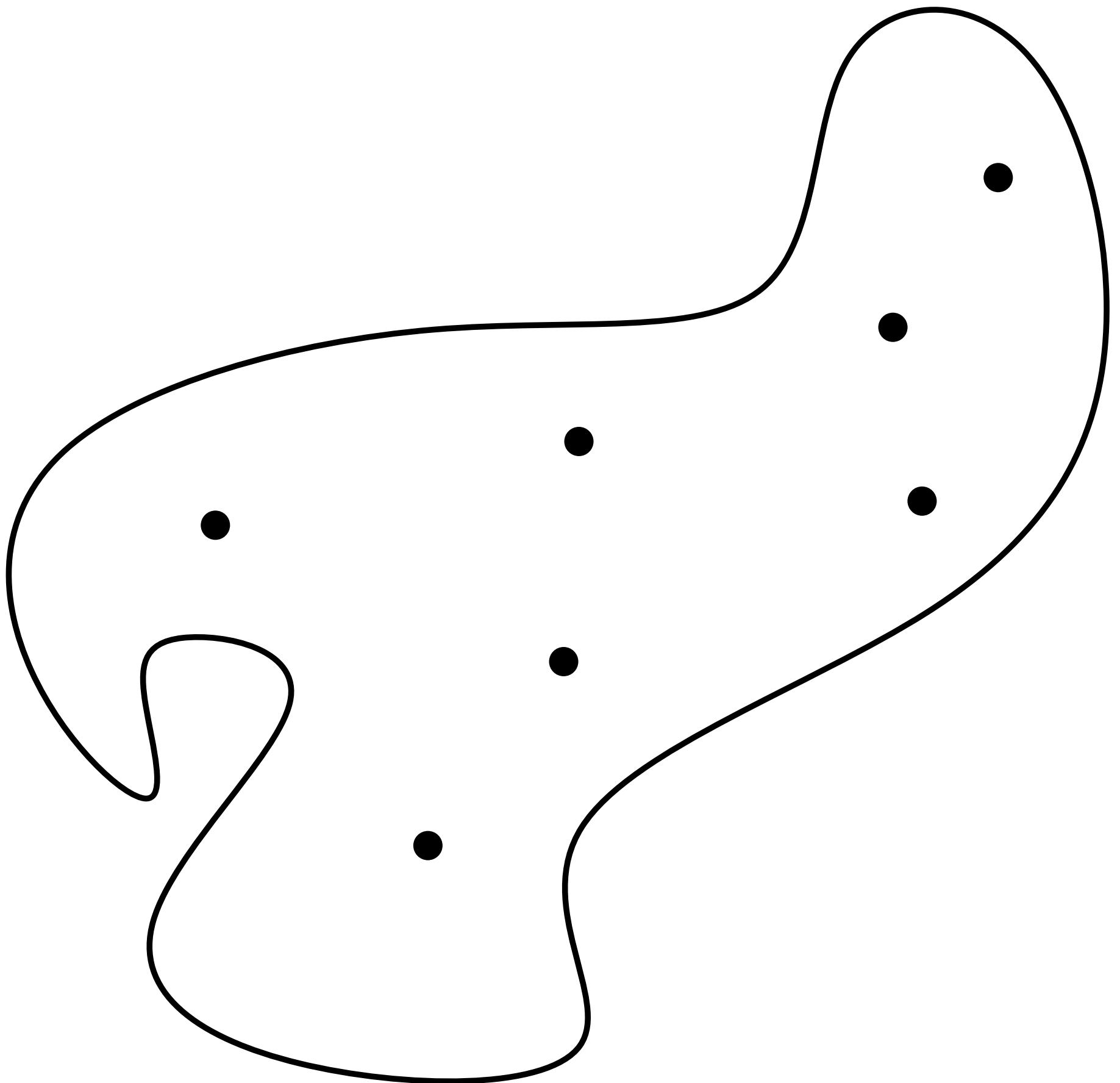


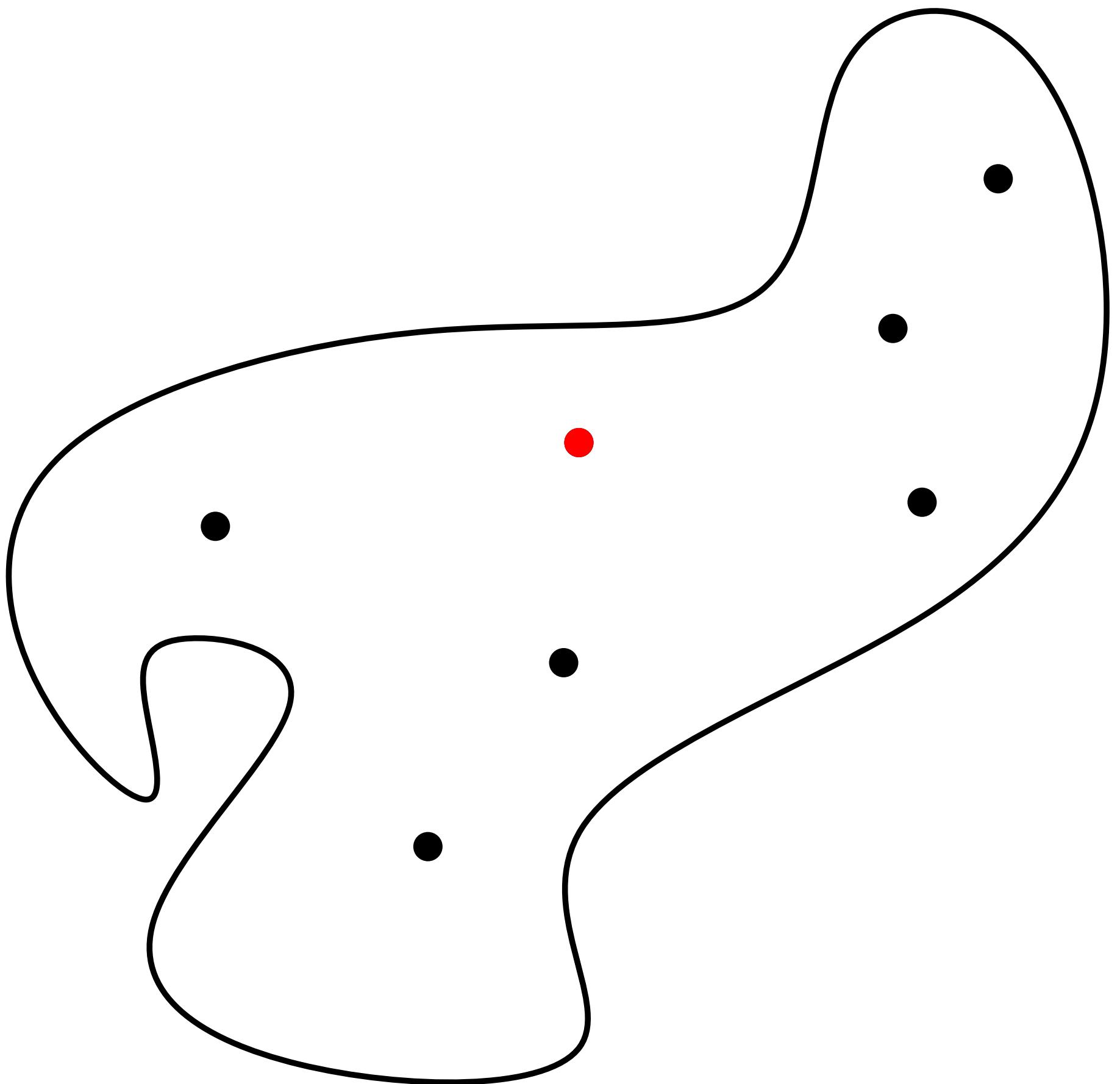


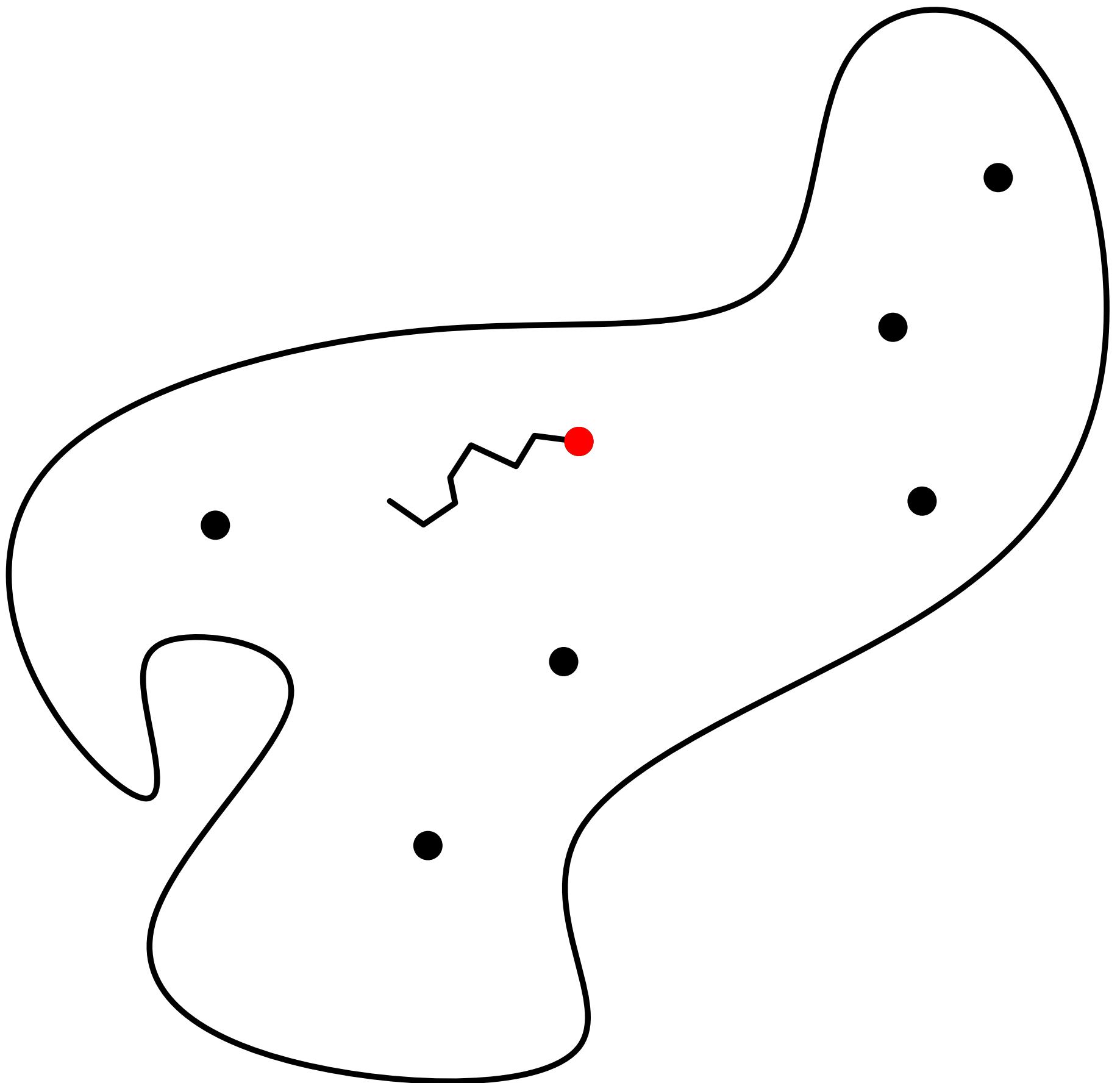


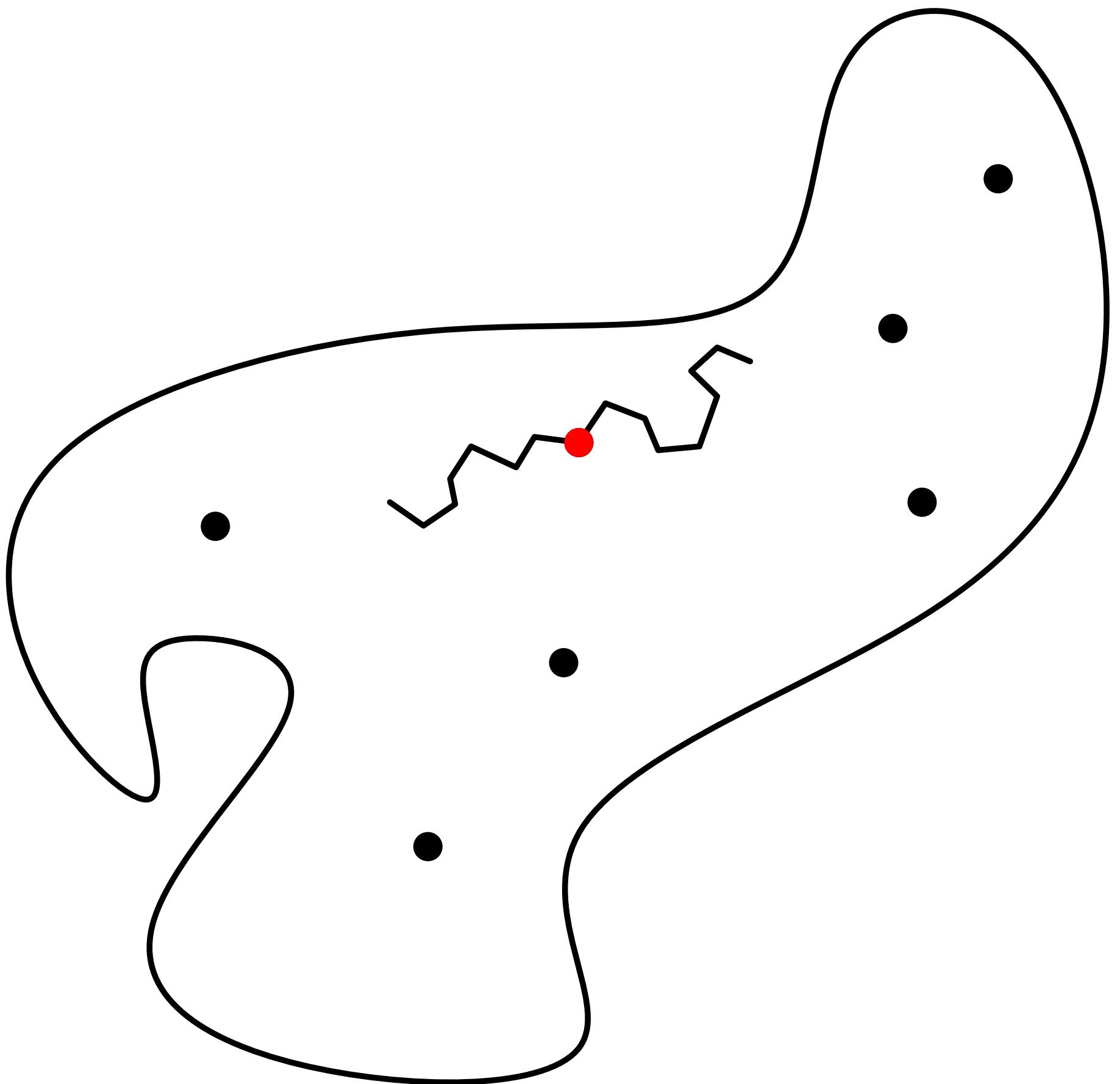


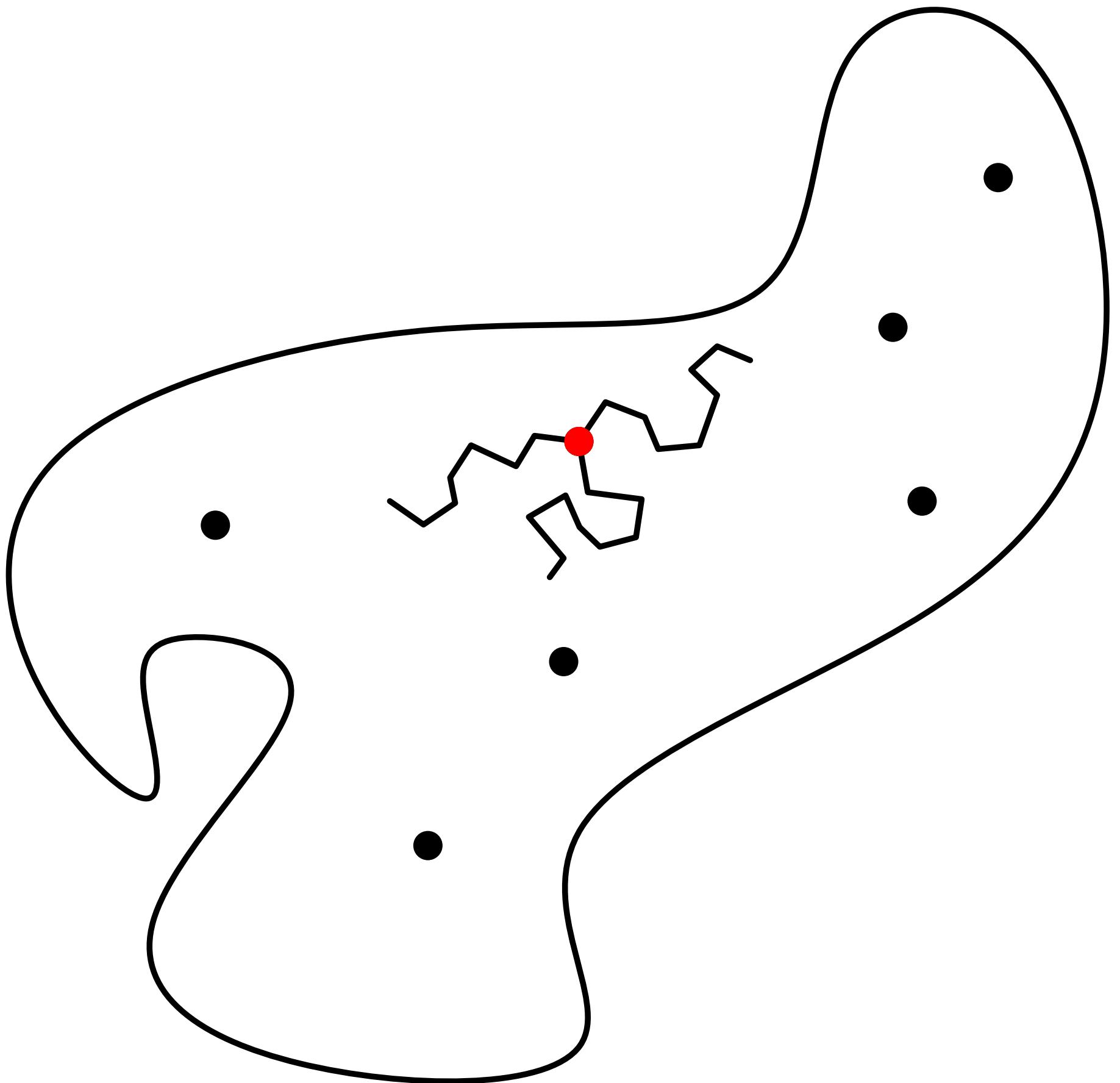


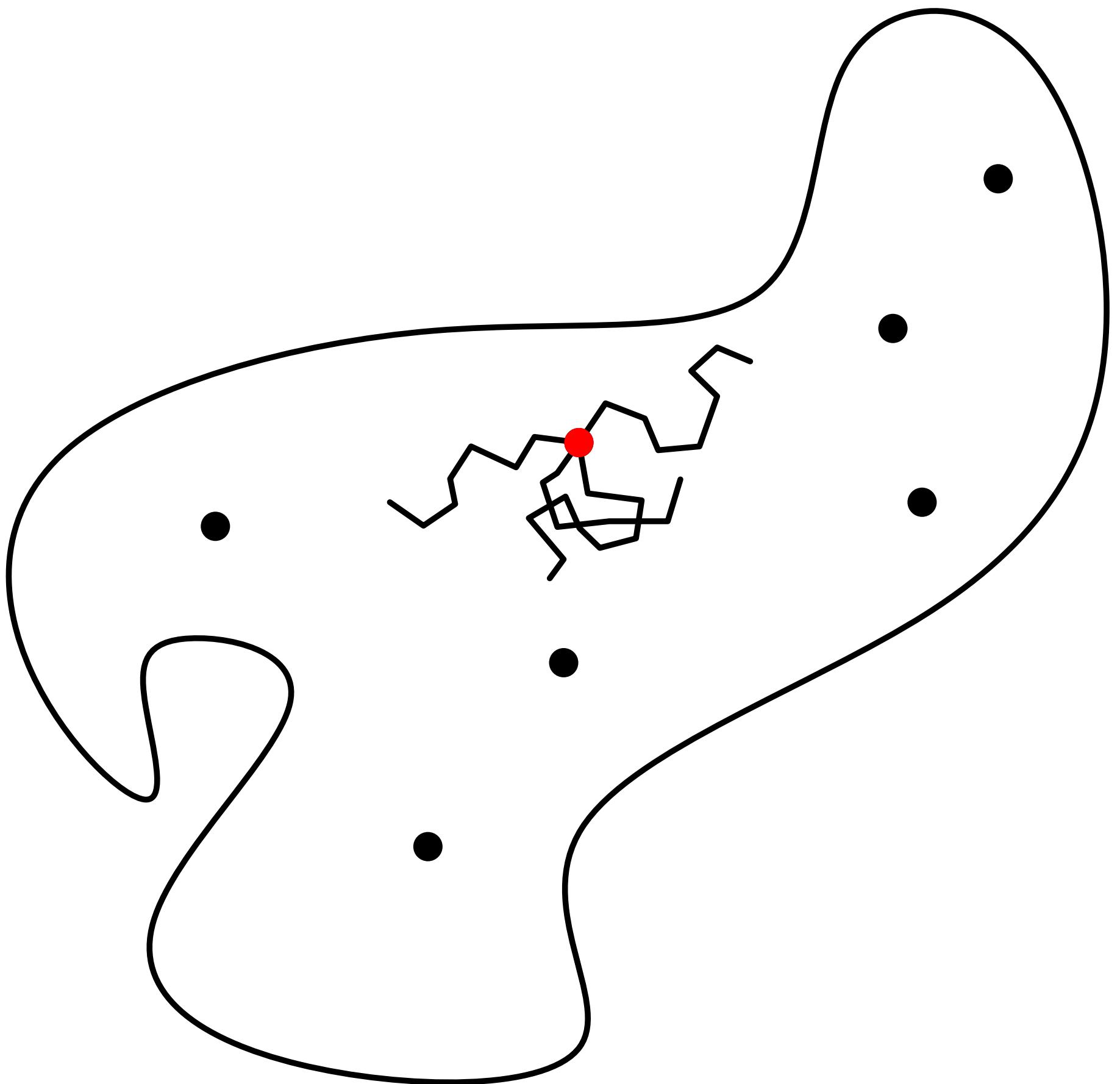




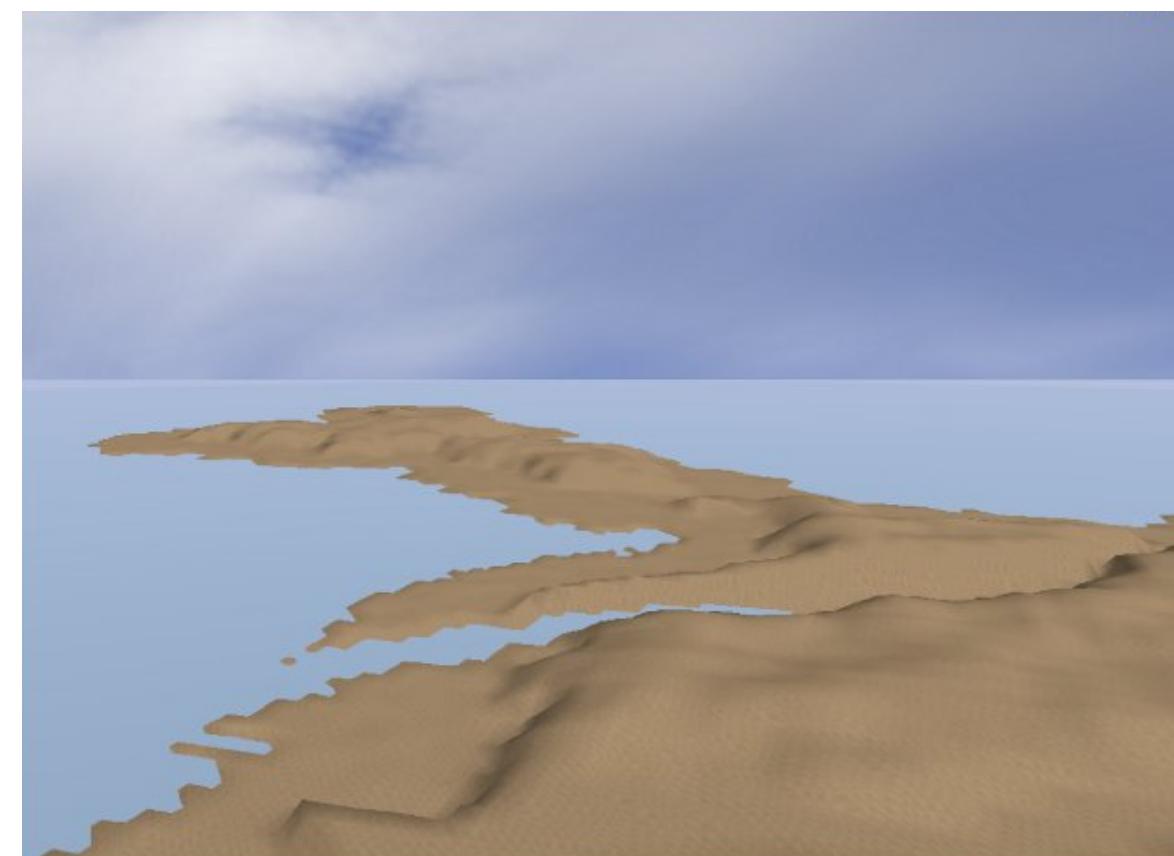
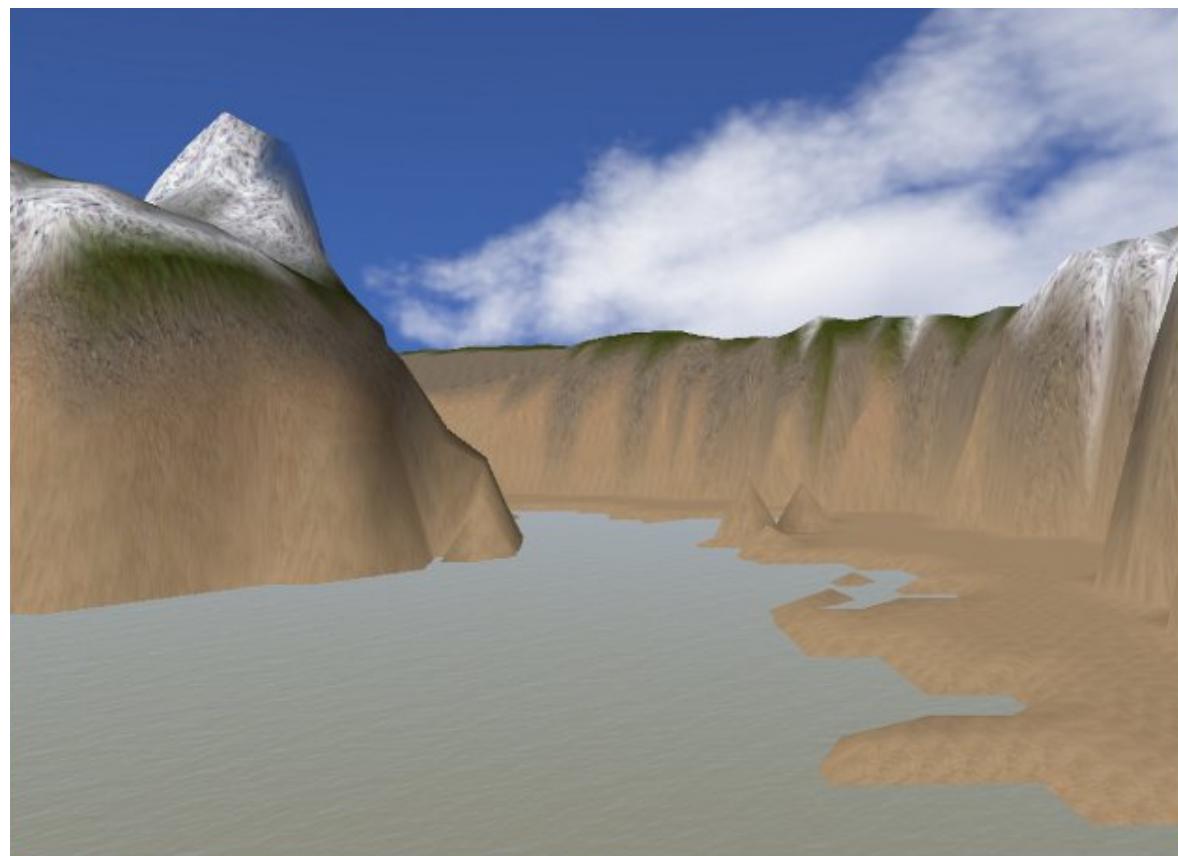
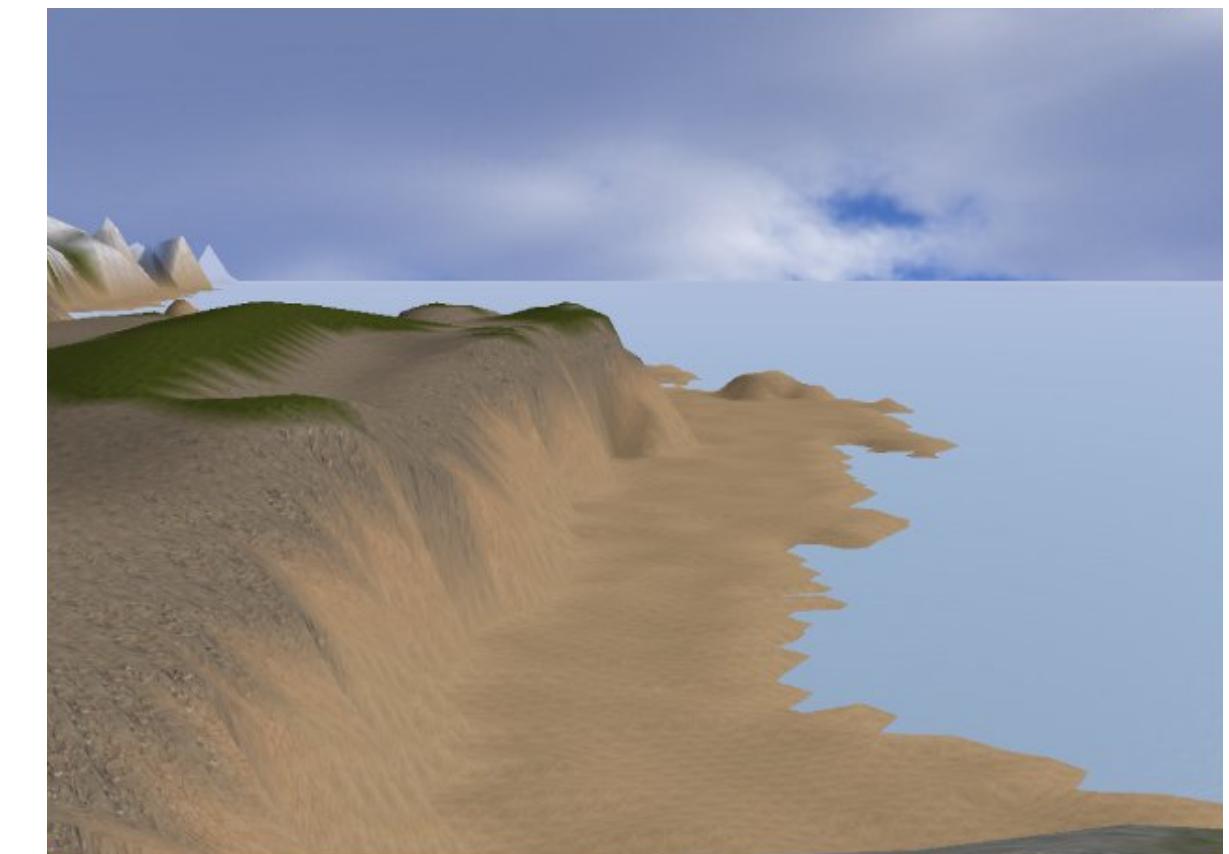
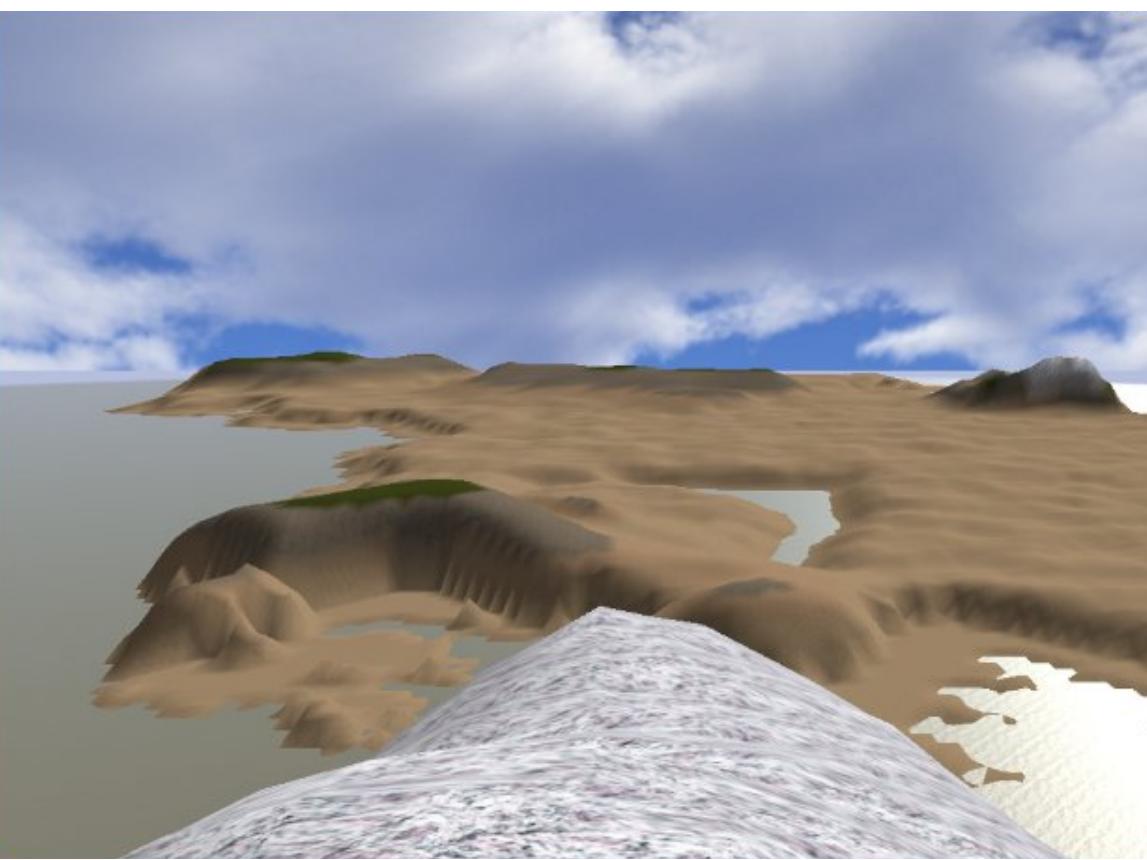








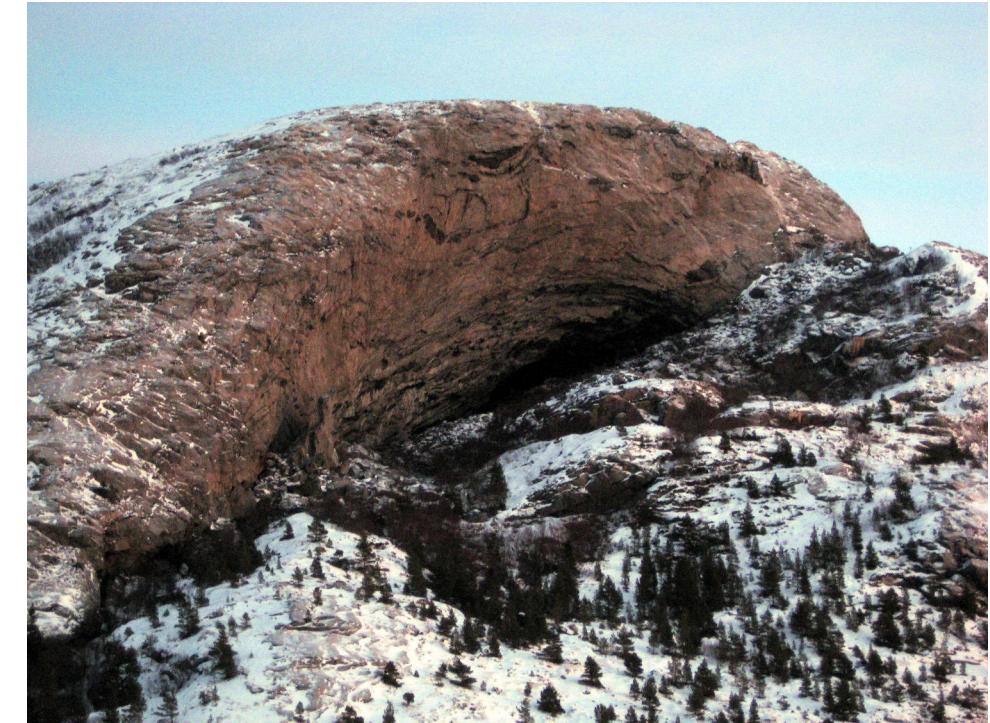
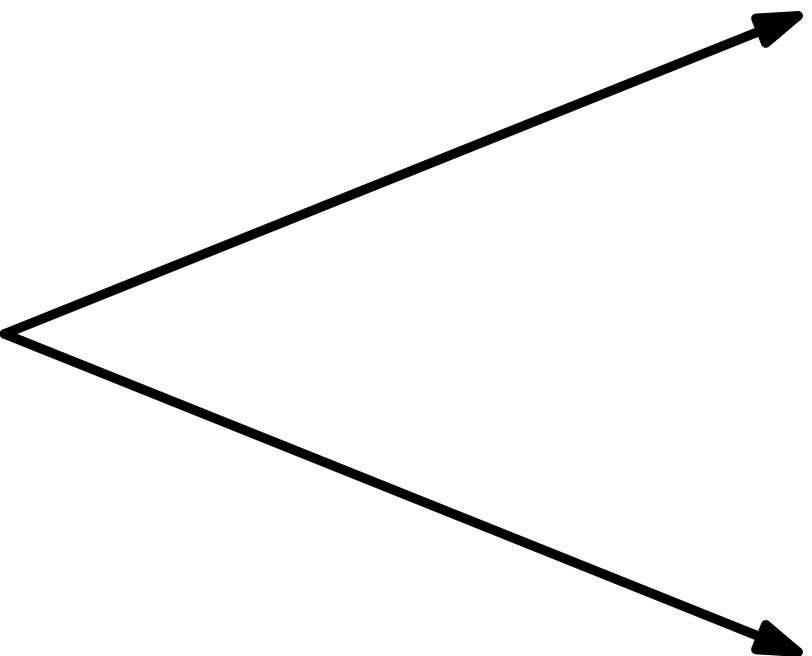
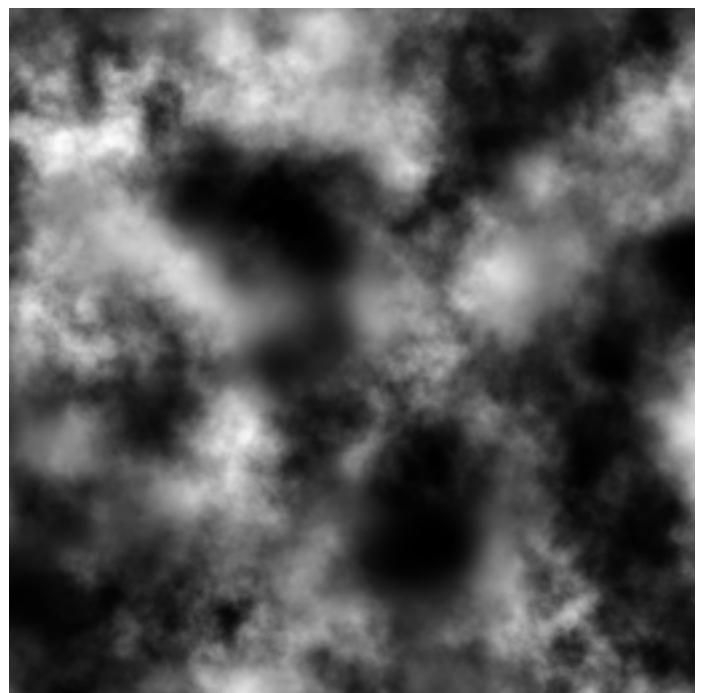




Assignment 4

Why is it important to restrict the operation of certain types of agents to certain phases of the terrain generation process?

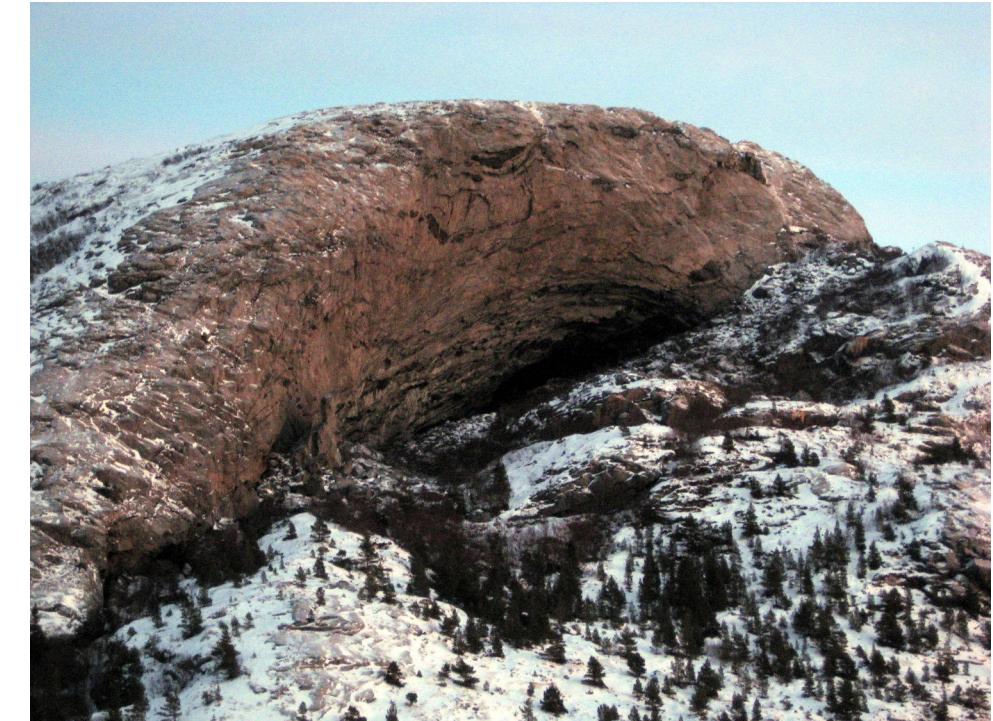
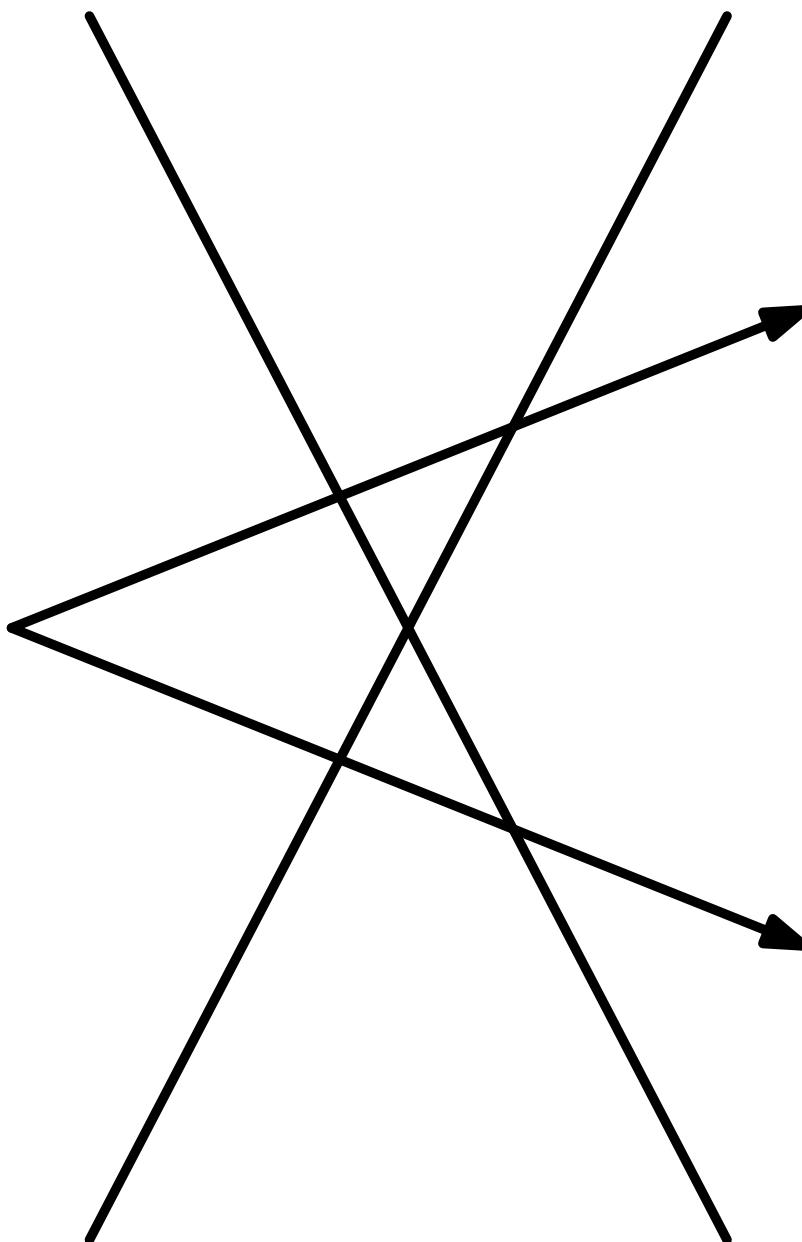
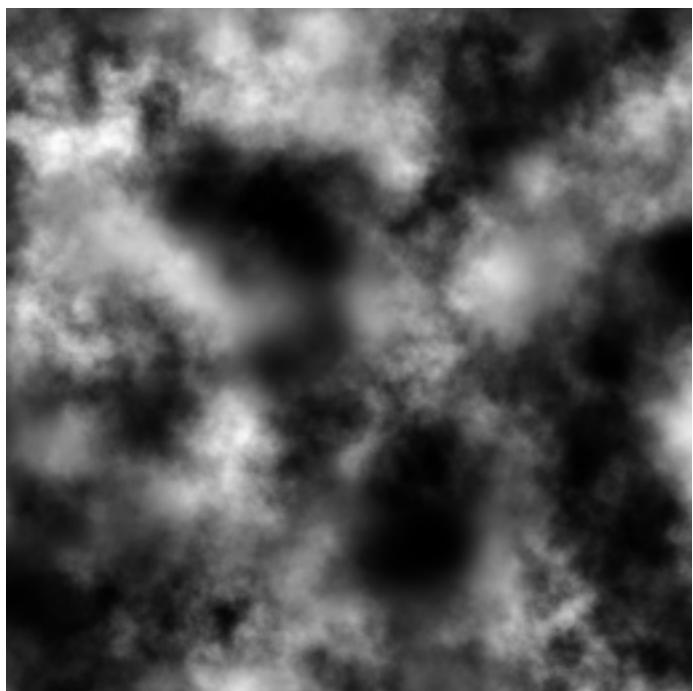
Voxel-based terrain generation



no.wikipedia.org/wiki/Hanshelleren



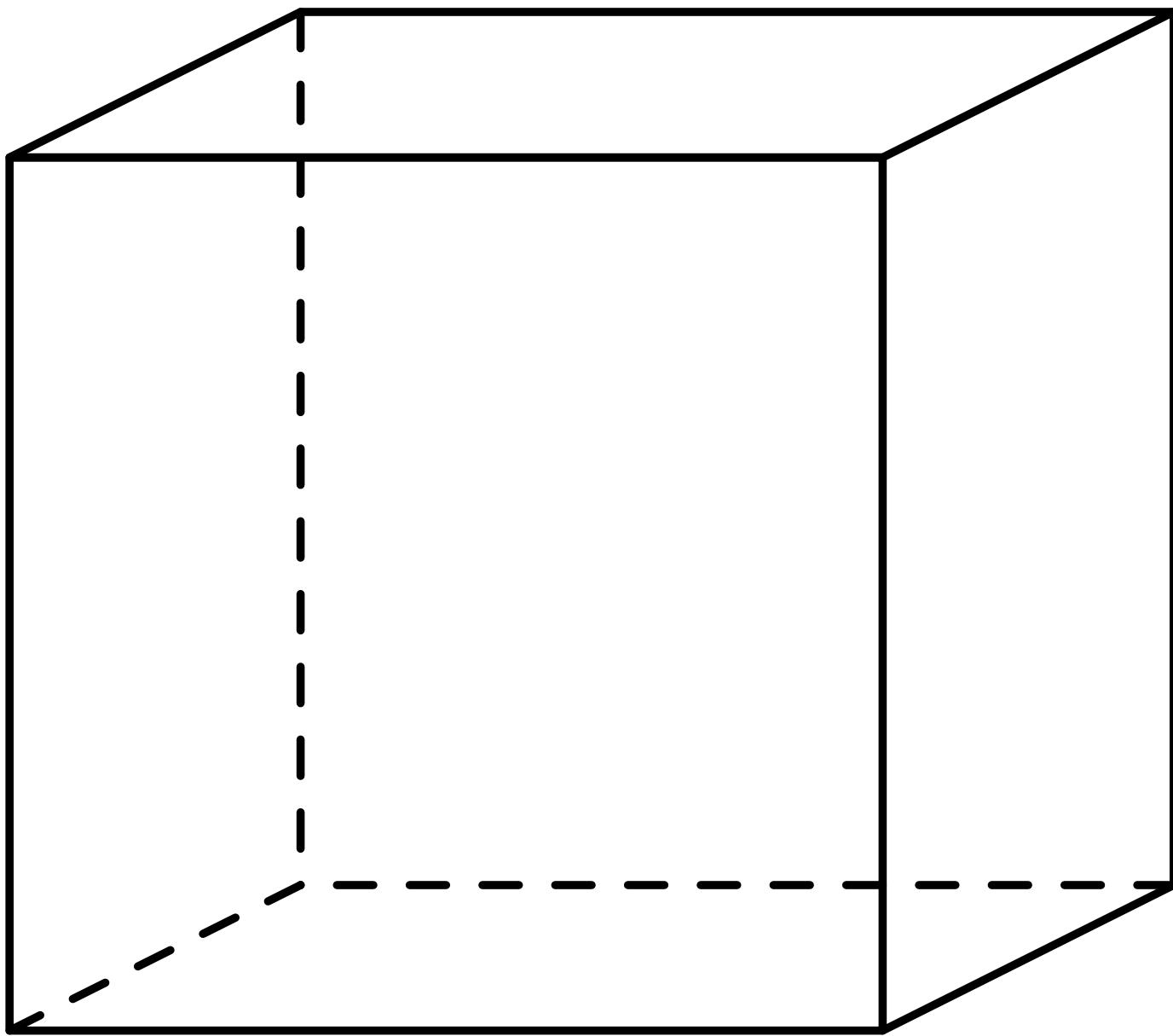
en.wikipedia.org/wiki/Kozarnika



no.wikipedia.org/wiki/Hanshelleren



en.wikipedia.org/wiki/Kozarnika





minecraft.gamepedia.com

Procedural feature generation for volumetric terrains using voxel grammars

Rahul Dey^{a,b,*}, Jason G. Doig^a, Christos Gatzidis^b



^aSony Interactive Entertainment Euro R&D, 13 Great Marlborough Street, London W1F 7HP, UK
^bBournemouth University, Poole House, Talbot Campus, Poole, Dorset BH12 5BB, UK

ARTICLE INFO

2010 MSC:

00-01

99-00

Keywords:

Procedural generation
Terrain
Voxels
Grammar

ABSTRACT

Terrain generation is a fundamental requirement of many computer graphics simulations, including computer games, flight simulators and environments in feature films. There has been a considerable amount of research in this domain, which ranges between fully automated and semi-automated methods. Voxel representations of 3D terrains can create rich features that are not found in other forms of terrain generation techniques, such as caves and overhangs. In this article, we introduce a semi-automated method of generating features for volumetric terrains using a rule-based procedural generation system. Features are generated by selecting subsets of a voxel grid as input symbols to a grammar, composed of user-created operators. This results in overhangs and caves generated from a set of simple rules. The feature generation runs on the CPU and the GPU is utilised to extract a robust mesh from the volumetric dataset.

1. Introduction

Generation of terrains can be a particularly important process when creating realistic representations of virtual worlds, as found in computer graphics simulations, feature films and computer games with outdoor environments. So far, there has been a considerable amount of research in this domain, which ranges between fully automated and semi-automated methods.

While it is now feasible to create massive virtual worlds, the tasks of designing the terrain, populating the world with content, and, finally, ensuring it does not feel empty or barren, continue to be very time consuming processes. Procedural content generation (PCG) has many applications and has proven valuable to designers due to its ability to algorithmically produce content such as the generation of textures, geometry and animations [1] so it can greatly improve the cost efficiency of populating a virtual environment. PCG will be used in this research to assist designers and shorten the length of time to create large scale landscapes.

Traditionally, terrains are defined by their surface details using a texture-based approach representing a top-down, two-dimensional view, called a *heightmap*. However, the details beneath the terrain surface have a significant impact on how the terrain is formed and its eventual appearance. This research uses volumetric data to represent terrain. This is important as it provides meaning to the details that are not visible to the user. Various factors, such as soil type and material density, govern how terrains are created in the real world. This can be modelled accurately when a voxel-based approach is utilized. A further

advantage of this approach is that both constructive and destructive methods to terrain creation can be adopted without being concerned about real-time polygon mesh editing, i.e. a surface can be extracted from the voxel data after the data has been constructed to the designer's liking.

This article proposes a procedural, voxel-based approach to assist users in the generation of key terrain features, such as overhangs and caves. The presented method expands the concept of *shape grammars* to a volumetric space and explains the process employed to create terrain features. We develop specific rulesets that are applied over a voxel dataset in order to create such features on the CPU. We also describe some good practices to be utilised when developing these rulesets. The final terrain mesh is generated at real-time frame rates by using our GPU-based surface nets algorithm. Furthermore, we present timings and memory usage from our results for the generation of the voxel data using different rulesets plus the performance statistics of our GPU surface extraction algorithm.

This research has been carried out in collaboration with Sony Interactive Entertainment Euro Research and Development (SIE Euro R &D), working with their proprietary game engine *PhyreEngine*™ [2]. We also thank NVIDIA Corporation for their hardware donation of a *Titan X* GPU.

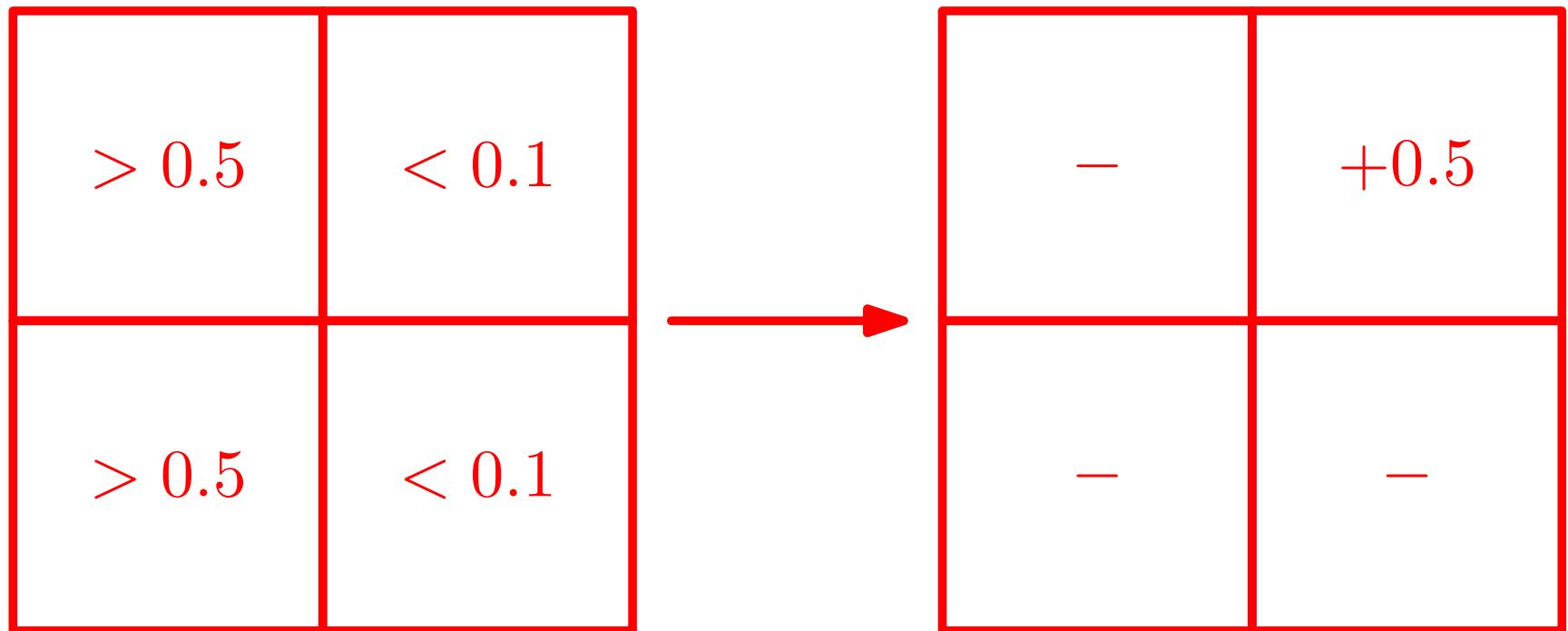
2. Related work

In this section previous work related to the formation of volumetric terrain is briefly reviewed. The proposed method involves aspects of

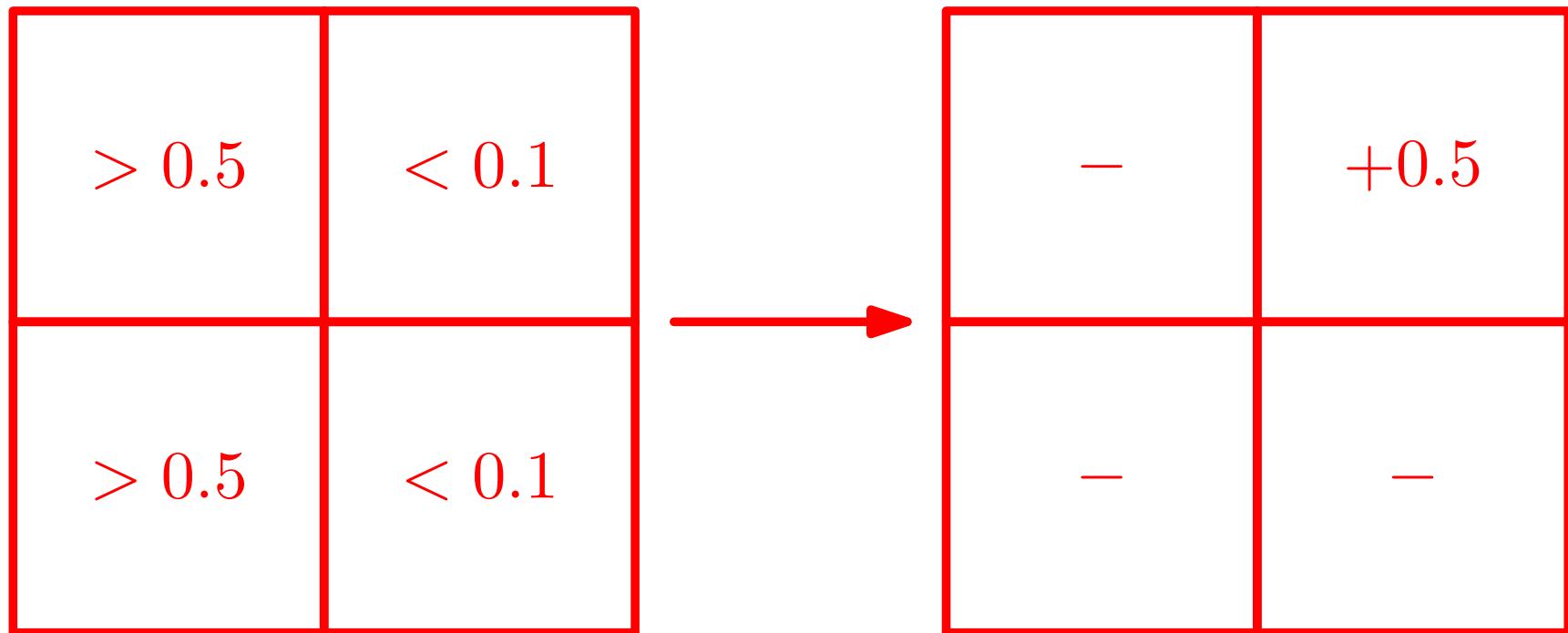
* Corresponding author at: Bournemouth University, Poole House, Talbot Campus, Poole, Dorset BH12 5BB, UK.
E-mail address: rahul.dey@bournemouth.ac.uk (R. Dey).

0	0	1	0
0	1	1	0
1	1	1	1
1	1	1	1

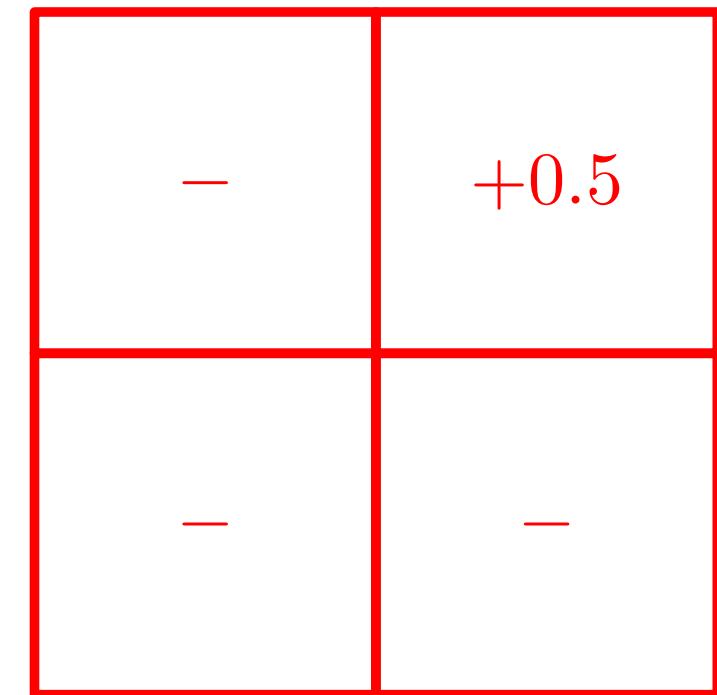
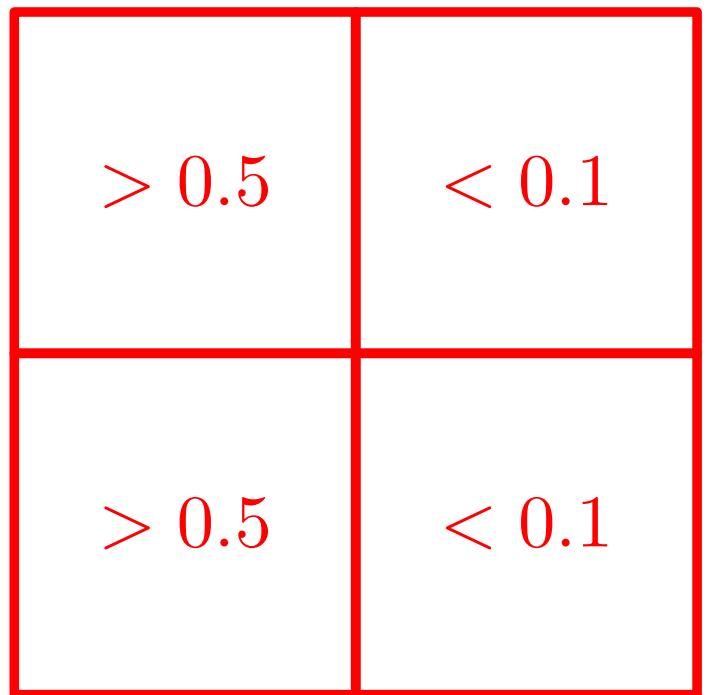
0	0	1	0
0	1	1	0
1	1	1	1
1	1	1	1



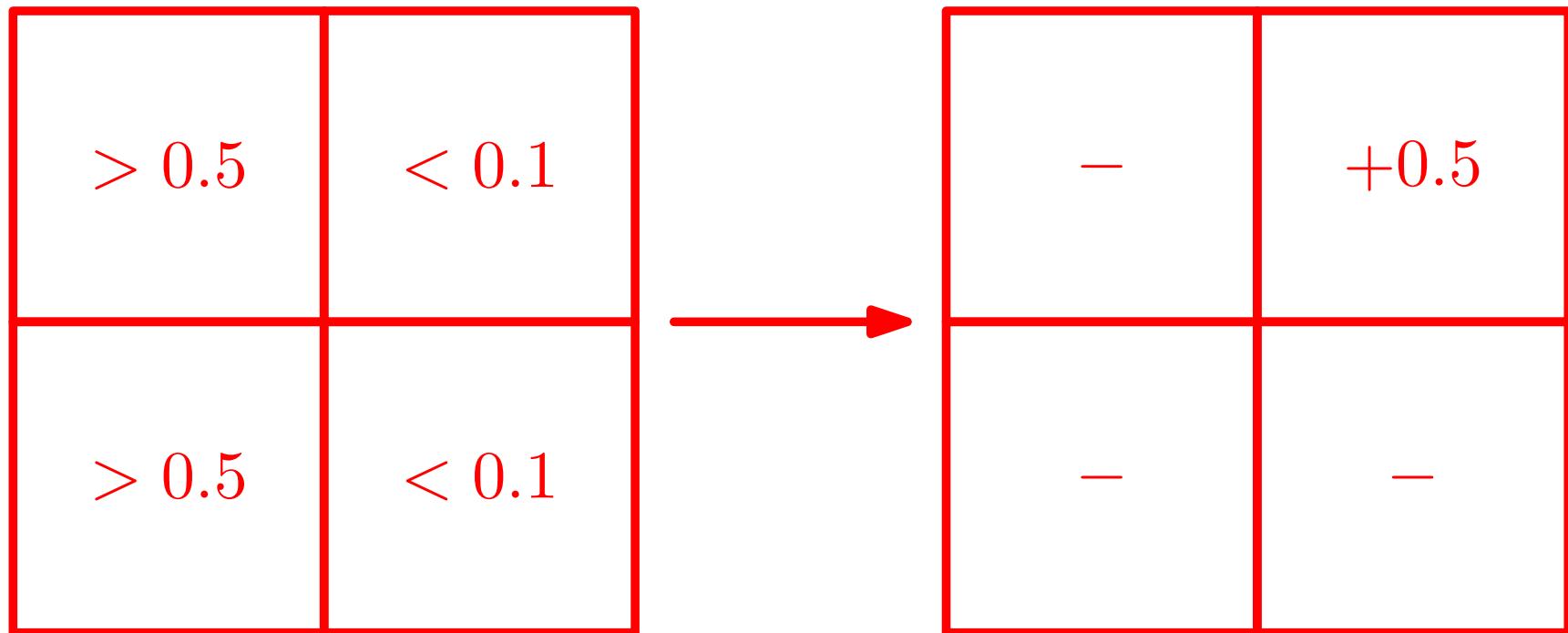
0	0	1	0
0	1	1	0
1	1	1	1
1	1	1	1

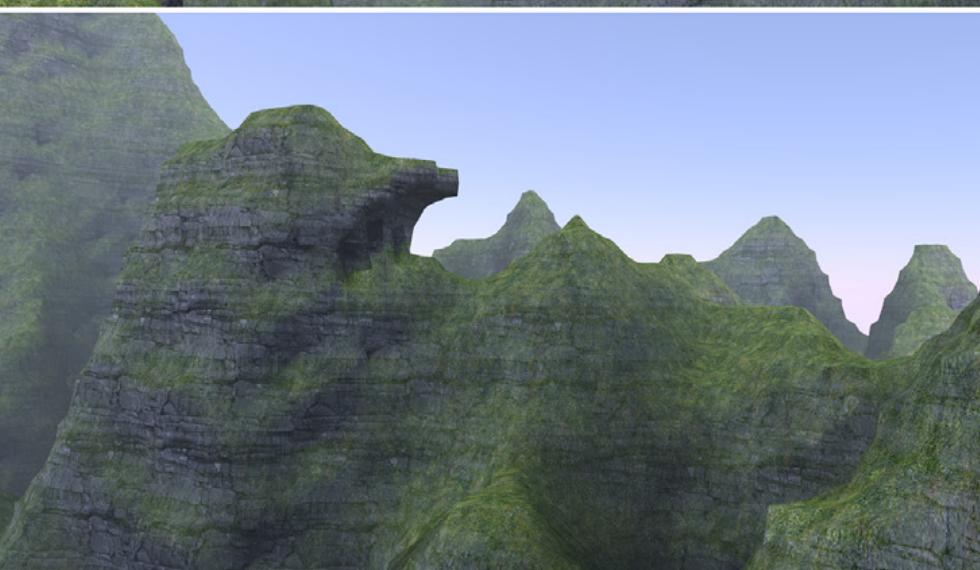


0	0	1	0
0	1	1	0
1	1	1	1
1	1	1	1



0	0	1	0.5
0	1	1	0
1	1	1	1
1	1	1	1





Assignment 5

If we add a cave agent to our agent-based system, in what phase should it run? Why?