"Images et géométrie discrète"
Final Exam (3h)

All exercises are independent. All material (lecture notes, slides) are allowed. Questions with $\star$ symbols may be more difficult and thus may give you extra points.

1 Image Processing

Let us consider a one-dimensional image $I : \mathbb{R} \to \mathbb{R}$. We apply the following filter:

$$I^*(x) = \frac{1}{W} \sum_{y \in \Omega_x} I(y) f(|I(y) - I(x)|) g(|y - x|), \tag{1}$$

with

- $W = \sum_{y \in \Omega_x} f(|I(y) - I(x)|) g(|y - x|)$;
- Ω_x is a window around x;
- f and g are decreasing functions $[0, +\infty] \to \mathbb{R}^+$.

Question 1 *Why do we need the factor $\frac{1}{W}$ in Eq. (1)?*

Question 2 *If $f(u) = e^{\frac{-u^2}{2\sigma}}$ and $g(u) = 1$, what kind of filter are we considering? And what about $g(u) = e^{\frac{-u^2}{2\sigma}}$ and $f(u) = 1$?*

Question 3 *If both f and g are Gaussian functions, discuss pros and cons of such filter in image processing. (You can use figures with various profiles for I to illustrate the different cases.)*

In the Image Processing literature, Eq. (1) is called a *bilateral filter*.

2 Randomized Digital Plane Recognition.

In the sequel, let us consider the following definition: a digital set $Z \subset \mathbb{Z}^3$ is a *naive digital plane* if and only if there exists a normal vector $N(a, b, c) \in \mathbb{Z}^3$ and a bound $\mu \in \mathbb{Z}$ such that:

$$\forall z \in Z, \ \mu \leq N \cdot z < \mu + \max(|a|, |b|, |c|) \tag{2}$$

(where $\cdot$ denote the scalar product).

We assume now that $0 \leq a \leq b < c$.

Question 4 *Show that in such case, any naive digital plane P is functional, i.e. for each pair $(z_1, z_2) \in \mathbb{Z}^2$, there is one and only one point z of coordinates (z_1, z_2, z_3) that belongs to P.*

Question 5 *Show that (2) implies that:*

$$\forall z \in Z, \begin{cases} N \cdot z \leq \mu \\ N \cdot (z + (0,0,1)) \geq \mu \end{cases} \tag{3}$$

Even if the converse is not true due to the large inequalities, we say that a digital set Z is a digital plane if the inequality set (3) is verified.

Question 6 *There exists a unique (Euclidean) plane passing through three digital points. Show that we can test whether another digital point lies BELOW, ON or ABOVE such a plane with integer-only computations.*

Question 7 *Detail a naive algorithm to decide if two digital set of points can be separated by an Euclidean plane.*

Let us now consider Algorithm 1 (which uses Algorithm 2). It is a randomized and recursive algorithm that checks whether two point sets are separable by a plane in expected linear-time.

We denote by $S = S^+ \cup S^-$ with S^- being the original point set and S^+ being the point set shifted by $(0, 0, 1)^T$. All the points of S are indexed from 1 to $|S|$. The idea consists in maintaining a separating plane while iterating over the points $s_i \in S$ for i from 1 to $|S|$. For each point s_i, three cases may occur:

- if it belongs to S^- (resp. S^+) and it is located BELOW (resp. ABOVE) or ON the current separating plane, there is nothing to do.
- Otherwise (lines 6-9 of algorithm 2):
  1. Either the two input sets are not separable by a plane at all,
  2. or there exists a separating plane passing through s_i.

In the aim of deciding between these last two alternatives, the set of possible separating planes is restricted to planes passing through s_i and the same algorithm is recursively called from 1 to i (line 9 of algorithm 2). At each recursive call, the set of possible separating planes is restricted so that the base case involves a unique plane passing by three given points and consists in checking whether it separates S^- from S^+ or not (lines 11-17 of algorithm 2).

Algorithm 1: isDigitalPlane(Z, p_1, p_2, p_3)

Input: $Z \subset \mathbb{Z}^3$, the digital set
$p_1, p_2, p_3 \in \mathbb{Z}^3$, three points characterizing a plane
Result: "true" if Z is a digital plane, "false" otherwise
Output: p_1, p_2, p_3, three points characterizing a separating plane if "true"
`// initialisation step`
Construct the set $S^- = Z$ and the set S^+ a copy of Z translated by $(0, 0, 1)$;
Construct the set of $S = S^- \cup S^+$;
Randomly permute the points of S ;
Initialize points $p_1 = p_2 = p_3 = (0, 0)$ (these points will be overwritten by *areLinearlySeparable*) ;
return areLinearlySeparable(S^-, S^+, S, $|S|$, p_1, p_2, p_3, 3) ;

Hint: to understand the behavior of Algorithm 1 and 2, you can consider a 2D example (separability test of two 2D point sets by an Euclidean line).

Question 8 *If Algorithm 1 terminates with a TRUE output, how to I recover the separating plane normal vector ?*

Algorithm 2: areLinearlySeparable(S^-, S^+, S, n, p_1, p_2, p_3, k)

Input: $S^-, S^+ \subset \mathbb{Z}^2$, the bottom and top point sets, $S = S^- \cup S^+$
n, number of points of S to process ($1 \leq n \leq |S|$)
$p_1, p_2, p_3 \in \mathbb{Z}^3$, three points characterizing a plane
k, number of variable points among $\{p_1, p_2, p_3\}$ ($0 \leq k \leq 3$)
Result: "true" if S^- and S^+ are separable by a plane, "false" otherwise
Output: p_1, p_2, p_3, three points characterizing a separating plane

```
areSeparable ← TRUE ;
if k > 0 then
    // we update points in {p1,p2,p3} which are ''free''
    for l from 1 to k do
        Initialize p_l with a point of S \ {p_j} for j > k ;
    i ← 1 ;
    while areSeparable and i < n do
        if (s_i ∈ S⁻ and s_i is strictly ABOVE the plane passing by p1,p2,p3)
        or (s_i ∈ S⁺ and s_i is strictly BELOW the plane passing by p1,p2,p3) then
            p_k ← s_i ;
            areSeparable ← areLinearlySeparable(S⁻, S⁺, S, i, p1, p2, p3, k − 1) ;
        i ← i + 1 ;
else
    i ← 1 ;
    while areSeparable and i < n do
        if (s_i ∈ S⁻ and s_i is strictly ABOVE the plane passing by p1,p2,p3)
        or (s_i ∈ S⁺ and s_i is strictly BELOW the plane passing by p1,p2,p3) then
            areSeparable ← FALSE ;
        i ← i + 1 ;
return areSeparable ;
```

Question 9 *Prove that Algorithm 1 terminates with a correct answer.*

Question $\star$ 10 *Using similar construction, how to change Algorithm 1 and 2 in order to decide if two digital sets in dimension 2 can be separated by an Euclidean circle ?*

This exercise has been inspired by:

- R. Seidel, "Small-dimensional linear programming and convex hulls made easy", Discrete & Computational Geometry, 6(1):423-434, 1991
- de Berg, Mark and Cheong, Otfried and van Kreveld, Marc and Overmars, Mark, "Computational Geometry: algorithms and applications,", Springer, 2000

The proof of the (expected) linear computational cost of Algorithm 1 is described in these articles.

3 Let's shoot some rays

Let us consider a digital set $Z \subset \mathbb{Z}^2$ which corresponds to the digitization of a convex object $\mathcal{X}$ (*i.e.*, $Z = \mathcal{X} \cap \mathbb{Z}^2$). We suppose that $\mathcal{X}$ has compact support and is thus included in a $[0, N]^2$ window. Objects Z and $\mathcal{X}$ are only given implicitly: given a point $p \in \mathbb{Z}^2$ (resp. a point $x \in \mathbb{R}^2$), we have an oracle to decide if p (resp. x) is inside Z (resp. $\mathcal{X}$) or not.
We consider a digital ray, denoted $dray(p, \vec{d})$ (with $p, \vec{d} \in \mathbb{Z}^2$), such that:

$$dray(p, \vec{d}) = ray(d, \vec{d}) \cap \mathbb{Z}^2$$

A digital ray is thus a 1-D lattice of points $\{q \,|\, \vec{Oq} = \vec{Op} + k \cdot \vec{d},\ k \in \mathbb{Z}^+\}$.

Question 11 *If $\vec{d}$ had irrational components, how many digital points would have been in $dray(p, \vec{d})$?*

Question 12 *If $p = (0, 0)$ and $\vec{d} = (a, b)^T$ ($a, b \in \mathbb{Z}^+$), how many digital points belongs to the digital ray $dray(p, \vec{d})$ in the $[0, N]^2$ domain ?*

We assume that $\vec{d} \in \mathbb{Z}^2$ and that the oracle on Z is $O(1)$.

Question 13 *Let p be a point inside Z, using only the oracle and arithmetical operators (no rounding operator allowed), propose an algorithm that returns the farthest point (from p) in $dray(p, \vec{d})$ which belongs to Z. Thanks to the convexity of $\mathcal{X}$, can you prove its correctness ? What is its complexity ?*

Let us consider Algorithm 3 which performs a sequence of digital shooting in a specific case where the object $\mathcal{X}$ is an half-plane.

Algorithm 3: GeometricalShooting(r)

Input: A rational number r
Result: A sequence of points S

Set $p_{-2} = (1, 0)$, $p_{-1} = (0, 1)$, $i = 0$;
Let $S = \emptyset$;
Let l be the line $y = rx$;
while *true* **do**
    Let $\vec{d} = \vec{Op_{i-1}}$;
    if $dray(p_{i-2}, \vec{d}) \cap l \neq \emptyset$ **then**
        Let p_i be the intersection point;
        $S = S \cup p_i$;
        return S;
    else
        Let p_i be the farthest point from p_{i-2} such that p_i and p_{i-1} lie on different side of l;
        $S = S \cup \{p_i\}$;
        $i = i + 1$;

Question 14 *What is the output of GeometricalShooting(r) for $r = \frac{5}{8}$? Draw the points S in grids of Figure 1.*

Question 15 *If $r = \frac{a}{b}$, what is the arithmetical interpretation of p_0 and the integer k from which p_0 has been obtained in $dray(p_{-2}, O\vec{p}_{-1})$?*

Question 16 *What are the arithmetical interpretations of points p_i ? What is the computational cost of* GeometricalShooting*(r) ?*

4 Digital Convex Hulls

We want to prove now the following statement:

Theorem 1 *Let C be a convex polygon (with vertices in $\mathbb{Z}^2$) and let D be its diameter. Then, C has at most $O(D^{2/3})$ vertices.*

Question 17 *If C has diameter D, what is the upper bound for its perimeter $P(C)$?*

Let $\{p_i\}$ be the sequence of vertices of C (counterclockwise for instance). We denote by $\vec{v_i}$ the vector $\vec{p_i p_{i+1}}$ (and let $V = \{v_i\}$). Clearly, if C has N vertices, we have

$$P(C) = \sum_{i=1}^{N} \|v_i\|$$

Let us consider a given edge length l. We define by W the subset of vectors of V having edge length greater or equal to l. I.e.:

$$W = \{v_i \,|\, v_i \in V \text{ and } \|v_i\| \geq l\}\,.$$

Question 18 *As a function of l and D, how many vectors are in W? (Hint: you can upper/lower bound $\sum_{v_i \in W} \|v_i\|$.)*

Question 19 *How many times a given vector v_i appears in V?*

Question 20 *Give an upper bound on the number of digital vectors having a length smaller than l.*

Question 21 *Ad a function of l and D, give an upper bound on the number of vertices of C.*

Question 22 *Finally, find the parameter l maximizing the upper bound given in the previous question and conclude with a proof of Theorem 1.*

Geometrical ray shooting described in Section 3 and Theorem 1 have been used by Sariel Har-Peled to construct an output sensitive digital convex hull computation algorithm.

Har-Peled, S. "An Output Sensitive Algorithm for Discrete Convex Hulls." Computational Geometry, vol.10,p.125–138, 1998.