Geometric Algorithms
Assignment 1

Laura Baakman (s1869140)

September 9, 2014

A

The Euclidean distance between the n-dimensional points $\boldsymbol{a}$ and $\boldsymbol{b}$ is defined as:

$$d(\boldsymbol{a}, \boldsymbol{b}) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2}. \tag{1}$$

My implementation of this formula is provided in Listing 1.
The length of the polygonal line created by drawing line segments between consecutive points from the list $[\boldsymbol{p_0}, \ldots, \boldsymbol{p_n}]$ is:

$$\sum_{i=0}^{n} d(\boldsymbol{p_i}, \boldsymbol{p_{i+1}}). \tag{2}$$

The first step of my implementation computes the list `pairs`, which contains all pairs of points between which the distance should be computed. Actually computing this distance, using the function `euclidean_distance` and summing the results gives the length of the requested line, see Listing 2.

Listing 1: An implementation of the Euclidean distance.

```
def euclidean_distance(a, b):
    """Compute the euclidean distance between the n-dimensional points
    a and b."""
    return(
        sqrt(
            sum(
                [(a_i - b_i)**2 for a_i, b_i in zip(a, b)]
            )
        )
    )
```

Listing 2: Compute the length of the polygonal path between consecutive points.

```
def length_of_connecting_path(points):
    """Compute the length of the polygonal line that connects
    consecutive points."""
    pairs = zip(points, points[1:])
    return(
        sum(
            [euclidean_distance(a, b) for (a, b) in pairs]
        )
    )
```

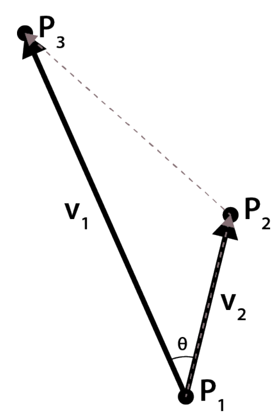

Figure 1: The three points $\boldsymbol{P_1}$, $\boldsymbol{P_2}$ and $\boldsymbol{P_3}$. The dashed line represents the path L from point $\boldsymbol{P_1}$, through $\boldsymbol{P_2}$ to $\boldsymbol{P_3}$. θ is the angle between the vectors $\boldsymbol{v_1} = \boldsymbol{P_3} - \boldsymbol{P_1}$ and $\boldsymbol{v_2} = \boldsymbol{P_2} - \boldsymbol{P_1}$

.

B

A general sketch of the situation is provided in Figure 1. To determine whether L makes a turn and in what direction we can use the angle θ. Since we are not interested in the exact angle, we only need to know whether θ is zero and if not, what its sign is.

Adding a z-coordinate of zero to the points $\boldsymbol{P_r}$ gives us the points $\boldsymbol{P_r}'$ with $r \in \{1, 2, 3\}$. The cross product of the vectors $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$ is defined as:

$$\boldsymbol{v_1} \times \boldsymbol{v_2} = \begin{pmatrix} 0 \\ 0 \\ q \end{pmatrix}, \qquad (3)$$

with $q = p1yp2x - p1xp2y - p1yp3x + p2yp3x + p1xp3y - p2xp3y$, where $p1y$ represents the second element of the point $\boldsymbol{p_1}$, the code used to derive this expression is presented in Listing 3. The sign of the variable q indicates what turn if any we made. If q is zero there was no turn, if q smaller than zero we

Listing 3: Generation of the formal expression.

```
p1 = {p1x, p1y, 0};
p2 = {p2x, p2y, 0};
p3 = {p3x, p3y, 0};
v1 = p3 - p1;
v2 = p2 - p1;
Cross[v1, v2]
```

made a right turn, if it is larger than zero a left turn.