DOI https://doi.org/10.15407/csc.2021.01.029
UDC 519.6

YE.V. VODOLAZSKIY, Senior Research Associate,
International Research and Training Centre of Information Technologies and Systems
of the NAS and MES of Ukraine, Glushkov ave., 40, Kyiv, 03187, Ukraine,
waterlaz@gmail.com

FRECHET SIMILARITY BETWEEN TWO AMBIGIOUSLY DEFINED POLYGONAL LINES

The paper considers the problem of comparing two polygonal lines that are not strictly defined. Instead, two sets of polygonal lines are given as sets of paths on two acyclic directed graphs. The problem is to determinewhether there exists a pair of lines each from its respective set such that the Frechet distance between them is not greater than a given number. An algorithm is given that solves the problem in $O(|E||R|)$ time, where E and R are the sets of edges in each graph respectively.

Keywords: *computational geometry, Frechet distance, computational complexity.*

Introduction

In pattern recognition practice the problem of comparing pairs of images is often reduced to comparing certain subsets of fields of view of images. Perhaps the most common example of such subsets are image contours [1—5], which are curves on the image without self-intersections. A natural problem of comparing such curves occurs and a typical and widely accepted metric for measuring the distance between curves is the Frechet metric [6]. There are quite a few publications regarding computing the exact or approximated Frechet distance between polygonal curves [6—8].

Unfortunately, the procedure of contour detection is not uniquely defined [1] and instead of one curve, the image may be represented by a set of curves (see Fig. 1). One of the curves from the set is the curve that gives the best description of the image. However, the selection of such curve is at least not obvious or even impossible without additional context.

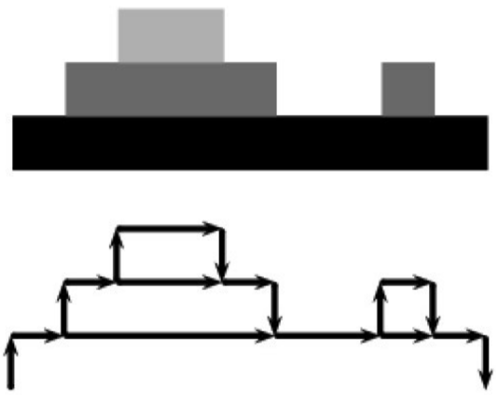


Fig. 1. An example of ambiguous detection of contours on a grayscale image with six possible contour curves

Problem definition

Let R^k be a linear space with k dimensions and a metric $d: R^k \times R^k \to R$ where $d(x_1, x_2)$ is the distance between points x_1 and x_2, $d(x_1, x_2) = \sqrt{(x_1 - x_2)^2}$.

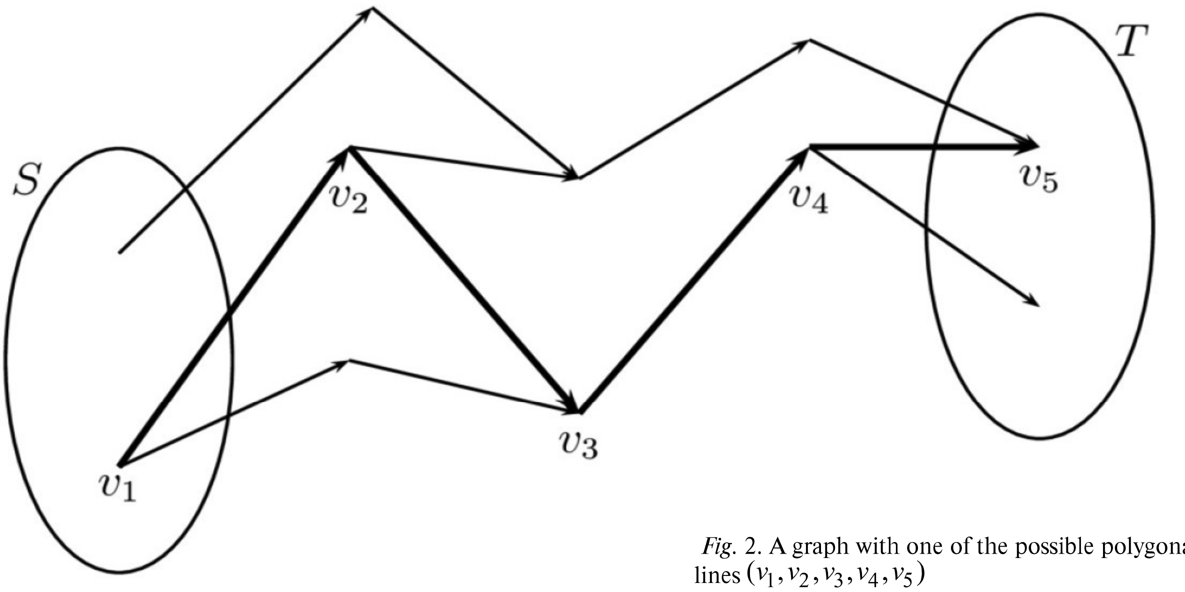


Fig. 2. A graph with one of the possible polygonal lines $(v_1, v_2, v_3, v_4, v_5)$

Definition 1. A polygonal line with m vertices $v_1, v_2, ..., v_m \in R^k$, is a continuous function $X:[0, m-1] \to R^k$ such that for any integer $i \in \{1, ..., m-1\}$ and any $\lambda \in [0,1]$ the equality $X(i+\lambda) = (1-\lambda)v_i + \lambda v_{i+1}$ holds.

For any polygonal line $X:[0, m-1] \to R^k$ denote R_x a set of such monotone nondecreasing functions $r:[0,1] \to [0, m-1]$ that $r(0)=0, r(1)=m-1$. Functions $r_X \in R_X$ are called reparametrizations of polygonal line $X:[0, m-1] \to R^k$, because both functions X and the composition $X(r_X(...))$ define the same set of points in R^k:

$$\{X(t) | t \in [0, m-1]\} = \{X(r_X(t)) | t \in [0,1]\}.$$

Definition 2. Frechet distance [6] between polygonal lines $X_1:[0, m-1] \to R^k$ and $X_1:[0, n-1] \to$ $\to R^k$ is the number:

$$\delta(X_1, X_2) \inf_{\substack{rX_1 \in R_{X_1} \\ rX_2 \in R_{X_2}}} = \max_{t \in [0\,1]} d(X_1(r_{X_1}(t)), X_2(r_{X2}(t))) .$$

For the sake of simplifying notation, X will also denote the set of points defined by the polygonal line X. In other words for a polygonal line X with m vertices

$$x \in X \Leftrightarrow \exists \alpha \in [0, m-1]: x = X(\alpha).$$

Let V be a finite set of points from a linear space R^k, such that $v \in R^k$ for $v \in V$. Let points V be vertices of some directed acyclic graph with edges $E \subset V \times V$. Moreover, suppose that two subsets $S \subset V$ and $T \subset V$ of vertices are given. We will name vertices S starting vertices and vertices T terminals.

The four sets $G = \langle V, S, T, E \rangle$ define a set of polygonal lines in the following way.

Definition 3. A sequence of vertices $(v_1, v_2, ..., v_m) \in V^m$ is called a path on the graph $G = \langle V, \underline{S, T, E} \rangle$ if $v_1 \in S, v_m \in T$ and $(v_i, v_{i+1}) \in E, i = \overline{1, m-1}$.

Every path $(v_1, v_2, ..., v_m) \in V^m$ defines a polygonal line with respect to Definition 1. Thus, every graph $\boldsymbol{G}$ defines a set of polygonal lines that we denote $\Xi(G)$ (see Figure 1). The set of polygonal lines $\Xi(G)$ is the set of lines that are formed by paths on the graph $\boldsymbol{G}$ that start at some starting vertex from S and end at some terminal from T. Since the graph G is acyclic, the length of each path is limited and therefore, the number of vertices in each polygonal line is also limited by the number of vertices in the graph. Moreover, the points from polygonal lines are partially ordered by $\preceq$ such that for a pair of points $x_1 \in R^k$ and $x_2 \in R^k$ the relation $x_1 \preceq x_2$ holds if and only if there is a path from x_1 to x_2.

Definition 4. Frechet distance between two sets of polygonal lines that are defined by graphs G_1 and G_2, is a number

$$\delta_G(G_1, G_2) \min_{X_1 \in \Xi(G_1)} = \min_{X_2 \in \Xi(G_2)} \delta(X_1, X_2),$$

where $\delta(X_1, X_2)$ is the Frechet distance between lines X_1 and X_2.

The paper solves the problem of determining whether for two given graphs G_1, G_2 and a number ε, the Frechet distance $\delta(G_1, G_2)$ between said graphs is not greater than the number ε.

In other words, the problem for a pair of graphs is to determine whether there is a pair of polygonal lines, each one from its respected graph, such that the Frechet distance between them is not greater than a given number.

Free-Space Diagram

The free-space diagram for a pair of graphs is a generalization of the free-space diagram for a pair of polygonal lines [6, 8, 9]. The free-space is a subset of pairs of close points on two graphs:

$$D^{\varepsilon} = \{(x, y) | x \in X, y \in Y, X \in \Xi(G_1), Y \in \Xi(G_2), d(x, y) \le \varepsilon\}.$$

Each point on the diagram corresponds to a pair of points from two graphs, such that the distance d between them is not greater than ε.

If graphs G_1 and G_2 define a single polygonal line each, the free-space diagram D^{ε} would just be a subset of a rectangle in R^2. Unfortunately, for a pair of arbitrary graphs G_1 and G_2 the free-space is not so simple and demonstrative. However, analyses of the free-space D^{ε} is greatly simplified by dividing it to cells that correspond to pairs of edges on each graph. For each pair of edges $e \in E$, $r \in R$ from two graphs $\langle V, S_1, T_1, E\rangle$ and $\langle U, S_2, T_2, R\rangle$ a cell from free-space is defined as

$$D_{er} = \{(x, y) | x \in e, y \in r, d(x, y) \le \varepsilon\}$$

$$D^{\varepsilon} = \bigcup_{\substack{e \in E \\ r \in R}} D_{er}.$$

Points from the free-space are also partially ordered:

$$(x_1, x_2) \preceq (y_1, y_2) \Leftrightarrow x_1 \preceq x_2,\ y_1 \preceq y_2.$$

Definition 5. A point $(x, y) \in D^{\varepsilon}$ is called reachable if the exists a monotone (in terms of relation $\preceq$) path from some starting point $(v_s, u_s) \in S_1 \times S_2$ to (x, y) on D^{ε}.

Now the question whether two graphs are ε-similar is reduced to the question of existence of a pair of reachable terminal points $(v_*, u_*) \in T_1 \times T_2$.

To formulate the algorithm we define sets of free points and sets of reachable points on the cell edges. For each vertex v of the first graph and each edge r of the second graph we define a set of free points on the left edge of the cell:

$$L_{vr} = \{\alpha \in [0,1] \| d(v, ((1 - \alpha)u_1 + \alpha u_2)) \le \varepsilon, u_1, u_2 = r\}.$$

For each edge e of the first graph and each vertex u of the second graph we define a set of free points on the bottom edge of the cell:

$$B_{eu} = \{\alpha \in [0,1] \| d(v, ((1 - \alpha)v_1 + \alpha v_2)) \le \varepsilon, v_1, v_2 = e\}.$$

The next lemma is a generalization of a corresponding lemma for a pair of polygonal lines[6].

Lemma 1. The sets L_{vr} for $v \in V$, $r \in R$ and the set B_{eu} for $e \in E$, $u \in U$ are closed convex sets.

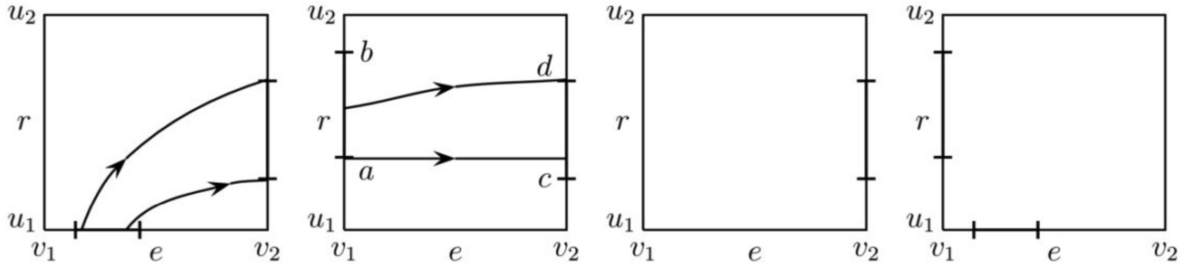
Proof. Indeed, the set L_{vr} defines a subset of points from linear segment u_1u_2 that are closer to a given point v than ε. In other words, it is a subset of such α, that $(u - ((1 - \alpha)v_1 + \alpha v))^2 \le \varepsilon^2$. By opening the brackets we obtain a quadratic inequality with non-negative coefficient near the term α^2. Therefore, the subset of α, that satisfy this inequality is a linear segment (convex and closed).

Subsets $L^*_{vr} \subseteq L_{vr}$ and $B^*_{eu} \subseteq B_{eu}$ are subsets of reachable points in terms of Definition 5 on the free-space. The subsets of reachable points are also linear segments, as the following lemma states.

Lemma 2. Subsets L^*_{vr} for $v \in V, r \in R$ and B^*_{eu} for B^*_{eu} *for* $e \in E, u \in U$ are closed and convex subsets.

Proof. Suppose that $L^*_{vr} \subseteq [0,1]$ is not convex. It means that there exists a triple $\alpha_1 \le \alpha_2 \le \alpha_3$ such that $\alpha_1, \alpha_3 \in L^*_{vr}$, but $\alpha_2 \notin L^*_{vr}$. Since $\alpha_1, \alpha_3 \in L^*_{vr}$, it is also valid that $\alpha_1, \alpha_3 \in L_{vr}$.. But since L_{vr} is convex, then $[\alpha_1, \alpha_3] \subset L_{vr}$.. Therefore, since the point α_1 is reachable, the point α_2 is also reachable because for the point α_2 there exists a monotone path that consists from two parts: a path to α_1 and a path from α_1 to α_2 (which is a straight line from free-space). Therefore, $\alpha_2 \in L^*_{vr}$.

The main idea of the algorithm is to sequentially compute reachable subsets on cell edges based on the already computed reachable subsets on other cell edges.



a *b* *c* *d*

Fig. 3. Construction of R^*_{er} for a pair of edges $e = v_1 v_2$ and $r = u_1 u_2$

Let us introduce additional notation. For a pair of edges $e = v_1 v_2 \in E$ and $r = u_1 u_2 \in R$ we define two sets:

$R^*_{er} \subseteq B_{eu_2}$ is a set of points on the right side of the cell D_{er}, that are reachable by a path that goes through edges *e* and *r*;

$T^*_{er} \subseteq L_{v_2 r}$ is a set of points on top side of the cell D_{er}, that are reachable by a path that goes through edges *e* and *r*.

Algorithm Description

Computations that are performed for each cell of the diagram are similar to the computations that are done on each cell for a pair of polygonal lines [6]. Nevertheless, we provide these computations in detail. For each cell it is necessary to compute the subset of reachable points on the top and right edges of the cell based on the reachable subsets on the bottom and left edges of the cell. We show how for each cell $D_{er}, e = v_1 v_2, r = u_1 u_2$ one can compute R^*_{er} and T^*_{er} based on $L^*_{v_1 r}$ and $B^*_{eu_1}$.

Intervals R^*_{er} are computed in the following way:

$$B^*_{eu_1} \neq \varnothing \quad \Rightarrow R^*_{er} = R_{er}, \qquad \text{(Fig. 3, } a\text{)};$$

$$\left.\begin{array}{l} B^*_{eu_1} = \varnothing, \\ L^*_{v_1 r} = [a, b], \\ R_{er} = [c, d], \end{array}\right\} \Rightarrow R^*_{er} = [\max\{a, c\}, d], \quad \text{(Fig. 3, } b\text{)};$$

$$\left.\begin{array}{l} B^*_{eu_1} = \varnothing, \\ L^*_{v_1 r} = \varnothing, \end{array}\right\} \Rightarrow R^*_{er} = \varnothing, \qquad \text{(Fig. 3, } c\text{)};$$

$$R_{er} = \varnothing \quad \Rightarrow R^*_{er} = \varnothing, \qquad \text{(Fig. 3, } d\text{)};$$

The intervals T^*_{er} are computed similarly:

$$L^*_{v_1 r} \neq \varnothing \quad \Rightarrow \quad T^*_{er} = T_{er};$$

$$\left.\begin{array}{l} L^*_{v_1 r} \neq \varnothing, \\ B^*_{eu_1} = [a, b], \\ T_{er} = [c, d], \end{array}\right\} \Rightarrow T^*_{er} = [\max\{a, c\}, d];$$

$$\left.\begin{array}{l} L^*_{v_1 r} \neq \varnothing, \\ B^*_{eu_1} = [a, b], \end{array}\right\} \Rightarrow T^*_{er} = \varnothing;$$

$$T_{er} = \varnothing \quad \Rightarrow T^*_{er} = \varnothing.$$

After computing R_{er} for all edges *e*, that enter vertex v_2, the subsets $L_{v_2 r}$ are computed.

$$L_{v_2 r} = \bigcup_{e \in V \times \{v_2\} \cap E} R_{er}.$$

Similarly, after computing T_{er} for all edges *r* that enter u_2, the subsets B_{eu_2} are computed.

$$B_{eu_2} = \bigcup_{r \in U \times \{u_2\} \cap R} T_{er}.$$

It takes constant time to perform computations for each cell. Since the diagram consists of $|E||R|$ cells the following theorem is valid.

Theorem 1. Determining ε-similarity of two graphs can be computed in $O(|E||R|)$ time, where E is the set of edges of one graph and R is the set of edges of the other.

Conclusions

The provided algorithm determines whether there is a pair of polygonal lines in the given graphs such that the Frechet distance between them is less than a given number. The time complexity of the algorithm is proportional to the product of the number of linear segments of two graphs. Note that the known algorithm's [6] complexity for a pair of polygonal lines is proportional to the product of the number of linear segments in the given lines. In a sense, the presented algorithm is a generalization of the known algorithm to graphs.

REFERENCES

1. *Matas J., Chum O., Urban M., Pajdla T.*, 2004. "Robust wide baseline stereo from maximally stable extremal regions", Image and Vision Computing, 22, pp. 761–767.
2. *Marot J., Caulier Y., Kuleschov A., Spinnler K., Bourennane S.*, 2008. "Contour detection for industrial image processing by means of level set methods", Advanced Concepts for Intelligent Vision Systems, pp. 655–663.
3. *Kass M., Witkin A., Terzopoulos D.*, 1988. "Snakes: Active contour models", IEEE Proc, on Computer Vision and Pattern Recognition, 1, pp. 321–331.
4. *Adamek T.,* O'Connor N., 2003. "Efficient contour-based shape representation and matching", Proceedings of the 5th ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR'03), pp. 138–143. [online] Available at: <https://core.ac.uk/download/pdf/147596517.pdf>.
5. *Marvaniya S.*, 2013. "Simultaneous Part-Decomposition and Contour Matching using Dynamic Programming", PhD thesis, Indian Institute of Technology. DOI: 10.13140/RG.2.2.22615.42400.
6. *Alt H., Godau M.*, 1995. "Computing the frechet distance between two polygonal curves", International Journal on Computational Geometry and Appication, 5 (1), pp. 75–91.
7. *Driemel A., Har-Peled S., Wenk C.*, 2012. "Approximating the frechet distance for realistic curves in near linear time", Discrete & Computational Geometry, 48, pp. 94–127. [online] Available at: <https://link.springer.com/article/10.1007/s00454-012-9402-z>.
8. *Schlesinger M., Vodolazskiy E., Yakovenko V.,* 2016. "Frechet similarity of closed polygonal curves", International Journal of Computational Geometry and Applications, 26 (1), pp. 53–66. DOI: 10.1142/S0218195916500035.
9. *Schlesinger M., Vodolazskiy E., Yakovenko V.*, 2014. "Recognizing the similarity of polygons in a strengthened hausdorff metric", Cybernetics and Systems Analysis, 3, pp. 174–187.

Received 10.03.2021

ЛІТЕРАТУРА

1. *Matas J., Chum O., Urban M., Pajdla T.* Robust wide baseline stereo from maximally stable extremal regions. Image and Vision Computing. 2004. 22. P. 761–767.
2. *Marot J., Caulier Y., Kuleschov A., Spinnler K., Bourennane S.* Contour detection for industrial image processing by means of level set methods. Advanced Concepts for Intelligent Vision Systems. 2008. P. 655–663.
3. *Kass M., Witkin A., Terzopoulos D.*, Snakes: Active contour models. IEEE Proc, on Computer Vision and Pattern Recognition. 1. 1988. P. 321–331.
4. *Adamek T., O'Connor N.* Efficient contour-based shape representation and matching. MIR'03 : Proceedings of the 5th ACM SIGMM International Workshop on Multimedia Information Retrieval. 2003. P. 138–143. URL: https://core.ac.uk/download/pdf/147596517.pdf.
5. *Marvaniya S.* Simultaneous Part-Decomposition and Contour Matching using Dynamic Programming. PhD thesis, Indian Institute of Technology. 2013. DOI: 10.13140/RG.2.2.22615.42400.
6. *Alt H., Godau M.* Computing the frechet distance between two polygonal curves. International Journal on Computational Geometry and Appication. 1995. 5 (1). P. 75–91.
7. *Driemel A., Har-Peled S., Wenk C.* Approximating the frechet distance for realistic curves in near linear time. Discrete & Computational Geometry. 2012. 48. P. 94–127. URL: https://link.springer.com/article/10.1007/s00454-012-9402-z.
8. *Schlesinger M., Vodolazskiy E., Yakovenko V.* Frechet similarity of closed polygonal curves. International Journal of Computational Geometry and Applications. 2016. 26 (1). P. 53–66. DOI: https://doi.org/10.1142/S0218195916500035.
9. *Schlesinger M., Vodolazskiy E., Yakovenko V.* Recognizing the similarity of polygons in a strengthened hausdorff metric. Cybernetics and Systems Analysis. 2014. 3. P. 174–187.

Надійшла 10.03.2021

Є.В. Водолазський, старший науковий співробітник, Міжнародний
науково-навчальний центр інформаційних технологій та систем НАН та МОН України,
03187, Київ, просп. Академіка Глушкова, Україна,
waterlaz@gmail.com

РОЗПІЗНАВАННЯ СХОЖОСТІ НЕОДНОЗНАЧНО ЗАДАНИХ ЛАМАНИХ ЛІНІЙ У МЕТРИЦІ ФРЕШЕ

Вступ. У практиці розпізнавання образів проблему порівняння пари зображень часто зводять до пошуку контурів на зображеннях, що є кривими без самоперетинів, та порівнянню цих кривих. Однією з найпоширеніших метрик для аналізу схожості таких кривих є метрика Фреше, обчисленню якої для пари ламаних ліній присвячено багато публікацій. На жаль, у більшості випадків виділення контурів на зображенні не є однозначним. Отже при розгляді пари зображень замість порівняння пари кривих необхідно аналізувати пару сукупностей кривих.

Мета статті. Необхідно розробити алгоритм, який би за двома множинами ламаних ліній та заданому числу визначав, чи існує в цих множинах така пара ламаних (по одній із кожної множини), що відстань Фреше між ними не перевищує заданого числа.

Результати. У статті наведено алгоритм, що як вхідні дані отримує пару орієнтованих ациклічних графів, вершинами яких є точки метричного простору, а ребрами — прямолінійні відрізки, що з'єднують вершини, та додатнє число. Таким чином, кожний шлях на графі задає ламану лінію, а граф у цілому задає множину ламаних ліній. На виході алгоритм дає відповідь, чи існує у двох графах така пара ламаних, що відстань Фреше між ними не перевищує заданого числа. Час роботи алгоритму пропорційний добутку кількості ребер одного графа на кількість ребер другого.

Висновки. Відомі алгоритми, що за парою ламаних визначають, чи перевищує відстань Фреше між ламаними задане додатнє число, потребують часу, який є пропорційним добутку кількості прямолінійних відрізків однієї ламаної на кількість прямолінійних відрізків іншої ламаної. Отже отриманий результат є узагальненням відомого алгоритму.

Ключові слова: *обчислювальна геометрія, метрика Фреше, алгоритмічна складність.*