Diagonal-flip distance algorithms of three type triangulations

Deqiang Wang, Xian Wang, Shaoxi Li
Institute of Nautical Science and Technology
Dalian Maritime University
Dalian 116026, P.R. China
dqwang@dlmu.edu.cn, hellen_wangdl@hotmail.com,
shaoxi_li@hotmail.com

Shaofang Zhang
Neusoft Denso Development Center
Neusoft Co., Ltd
Dalian 116023, P.R. China
zhang.shf@neusoft.com

***Abstract*—In this paper we study the diagonal flipping problem in three special type triangulations of n ($n > 7$) vertex convex polygons (and rotations in three type binary trees). By labelling vertices of the triangulations, we propose six linear time algorithms for computing diagonal-flip distances in these three type triangulations (and rotation distance in these three type binary trees).**

Index Terms - triangulation; diagonal-flip distance; binary tree; rotation distance.

I. INTRODUCTION

A diagonal-flip transformation is an operation that converts one triangulation into another by removing a diagonal in the triangulation and adding the diagonal that subdivides the resulting quadrilateral in the opposite way. The dual edge $dual(e)$ is an edge that appears if we flip the edge e. A rotation in a binary tree is a local restructuring of the tree that changes the position of an internal node and one of its children while the symmetric order in the tree is preserved (see Fig.1). Culik and Wood defined in 1982 the rotation distance between a pair of binary trees as the minimum number of rotations needed to convert one tree into the other [4]. There is a well-known explicit bijection between binary trees and triangulations [10], thus a system that is isomorphic to triangulations of a convex polygon related by the diagonal-flip transformation is that of binary trees related by rotations. Therefore, diagonal-flip distance of triangulations and rotation distance of binary trees are equivalent.

Much research have been worked on diagonal-flip (rotation) distance and some upper bounds of these distances have been exhibited ([1], [5], [6]). Some authors approach the problem by limiting the reshaping primitive to a restricted version of the general rotation operation ([2], [3], [7], [9], [12]). Obviously these restricted rotation distances will be bounded below by the usual rotation distance. However, there is an open problem whether these distances can be computed in polynomial time. In Ref.[8], Lucas presented a quadratic time algorithm for computing the rotation distance between binary trees of restricted form (the original tree and the destination tree are both degenerated tree, and the destination tree has at most one zig-zag pair of edges), and gave the exact rotation distance $2n-2+|\Lambda|+\eta$ (where Λ is the longest common subsequence between the original and destination tree, $|\Lambda|$ is the length of Λ, η has values 0 or 1 related to Λ).

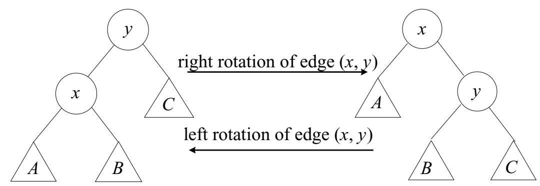

Fig. 1. A rotation in a binary tree

In this paper, by labeling vertices of three special types of triangulations, we present polynomial time algorithms to find the exact diagonal-flip distance between them, then, according to the well-known explicit bijection between triangulations and binary trees, we conclude the rotation distance between binary trees corresponding to the special triangulations.

II. DEFINITIONS AND PRELIMINARIES

Denote P_n as an n ($n > 7$) vertex convex polygon, and $\mathcal{T}_n$ as the set of all triangulations of P_n. Let v_i be a vertex of P_n ($i \in \{1, 2, \cdots, n\}$).

(1) Connect $n-3$ vertices that are not adjacent to v_i to form a triangulation $T_{i,n-3}$ (see Fig.2(a)). Denote $\mathcal{L}_1 = \{T_{i,n-3} | i \in \{1, 2, \cdots, n\}\}$, and call $\mathcal{L}_1$ to be the type-I triangulation.

(2) Connect $n-4$ vertices that are not adjacent to v_i in succession by counterclockwise (clockwise) direction, and connect the $(n-4)$th vertex to the first vertex that are not adjacent to it by counterclockwise (clockwise) direction, in this way, form the triangulation $T_{i,n-4}$ ($T'_{i,n-4}$), see Fig.2(b) (see Fig.2(c)). Denote

$$\mathcal{L}_2 = \{T_{i,n-4} | i \in \{1, 2, \cdots, n\}\},$$

$$(\mathcal{L}_3 = \{T'_{i,n-4} | i \in \{1, 2, \cdots, n\}\}),$$

and call $\mathcal{L}_2$ ($\mathcal{L}_3$) the type-II triangulation (the type-III triangulation).

Let $T \in \mathcal{T}_n$ and v be a vertex of T. The diagonal degree of v is the number of diagonals incident to v, denoted by $Dia\_deg(v)$. For $T_1, T_2 \in \mathcal{T}_n$, the diagonal-flip sequence is

978-0-7695-3336-0/08 $25.00 © 2008 IEEE
DOI 10.1109/CSSE.2008.337

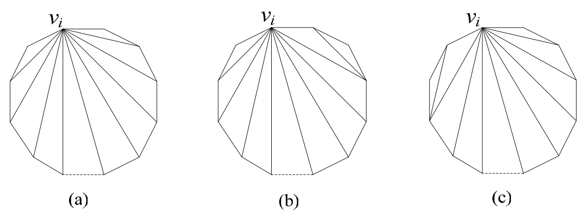

Fig. 2. Three type triangulations of convex polygons

formed by all the flipped diagonals which can convert T_1 into T_2, denoted by P_{T_1,T_2}. The length of P_{T_1,T_2} is denoted by $|P_{T_1,T_2}|$. It is easy to see that P_{T_1,T_2} is not unique.

The diagonal-flip distance (DFD for short and denoted by $Dist(T_1,T_2)$) between T_1 and T_2 is the minimum number of diagonal-flip transformations needed to convert T_1 into T_2. So, $Dist(T_1,T_2) = \min\{|P_{T_1,T_2}|\}$. There are following results about DFD:

Lemma 1([11]). *$\forall T_1, T_2 \in \mathcal{T}_n$, if T_1 and T_2 have a common diagonal, then this diagonal is not included in the shortest diagonal-flip sequence from T_1 to T_2.*

Lemma 2([11]). *$\forall T_1, T_2 \in \mathcal{T}_n$, if it is possible to flip one diagonal of T_1 creating T' so that T' has one more common diagonal with T_2 than T_1 does, then there exists a shortest diagonal-flip sequence from T_1 to T_2 in which the first flip creates T_1.*

From Lemma 2, we can get

$$Dist(T_1,T_2) = Dist(T',T_2) + 1.$$

Lemma 3([8], [11]). *$\forall T_1, T_2 \in \mathcal{T}_n$, if T_1 and T_2 share a common diagonal, and the diagonal splits T_1 (T_2) into two sub-triangulations T_1^1 and T_1^2 (T_2^1 and T_2^2) in such a way that T_1^1 and T_2^1 (T_1^2 and T_2^2) have the same vertices. Then we have the following formula:*

$$Dist(T_1,T_2) = Dist(T_1^1,T_2^1) + Dist(T_2^2,T_2^2).$$

III. Diagonal-flip distance algorithms

Let T_0, T_f be the original and the destination triangulations, respectively. The n vertices of T_0 are labelled counterclockwise with $1, 2, \cdots, n$ beginning from the vertex with maximum degree. The n vertices of T_f are labelled according to the vertices of T_0. And we use (i,j) to denote the diagonal of two unadjacent vertices i and j. Suppose that all vertices of a polygon can be both clockwise and counterclockwise travelled from each vertex.

A. DFD Algorithm in Type-I Triangulations

Algorithm 1 DFD in type-I triangulations

Input: $T_0, T_f \in L_1$
Output: $Dist(T_0,T_f)$
Step 1. In T_0, seek the maximum degree vertex and label n vertices;
Step 2. Label n vertices of T_f;
Step 3. $m =$ the label of the maximum degree vertex in T_f;
Step 4. $j = 0$;
Step 5. Flip diagonals from T_0:
    Case $m = 2$: For $i = 3$ To $n-1$
        Flip $(1,i)$, $j++$;
    Case $m = 3$: For $i = 4$ To $n-1$
        Flip $(1,i)$, $j++$;
    Case $m = 4, 5, \cdots, n-1$:
        For $i = m-1$ To 3
            Flip $(1,i)$, $j++$;
        For $i = m+1$ To $n-1$
            Flip $(1,i)$, $j++$;
    Case $m = n$: For $i = n-1$ To 3
        Flip $(1,i)$, $j++$;
Step 6. Output j;

we can gain degrees of every vertex from the triangulating of P (if a diagonal is formed by two vertices, then the degree of the two vertices are both added 1). In Step 1, we can find the vertex with maximum degree $n-3$ by travelling at most n vertices, which takes $O(n)$ time; labelling T_0 also takes $O(n)$ time, and we get $Dia\_deg(1) = n-3$, $Dia\_deg(2) = 0$, $Dia\_deg(n) = 0$, and the other vertices are all with diagonal degree 1. In Step 2, labelling T_f also takes $O(n)$ time. In Step 5, we divide m ($\in \{2, 3, \cdots, n\}$) into following cases

(1) if $m = 2$ or $m = n$, from Lemma 2, the flip distance is minimal: $Dist(T_0,T_f) = n-3$;

(2) if $m = 3$ or $m = 4, 5, \cdots, n-1$, from Lemma 1, Lemma 2 and Lemma 3, the flip distance is minimal: $Dist(T_0,T_f) = n-4$.

Obviously, each case in Step 2 can be done in $O(n)$ time. So we reach the result:

Theorem 1. *The time complexity of Algorithm 1 is $O(n)$, and the diagonal-flip distance in type-I triangulations is $n-4$ or $n-3$.*

B. DFD Algorithm in Type-II Triangulations

Algorithm 2. DFD in type-II triangulations

Input: $T_0, T_f \in L_2$
Output: $Dist(T_0,T_f)$
Step 1. In T_0, seek the maximum degree vertex and label n vertices;
Step 2. Label the n vertices of T_f;
Step 3. $m =$ the label of the maximum degree vertex in T_f;
Step 4. $j = 0$;
Step 5. Flip diagonals from T_0:
    Case $m = 2$: For $i = 3$ To $n-3$
            { Flip $(1,i)$, $j++$; }
        Flip $(n-2,n)$, $j++$;
        Flip $(1,n-2)$, $j++$;
    Case $m = 3$: For $i = 4$ To $n-2$
            { Flip $(1,i)$, $j++$;}
        Flip $(n-2,n)$, $j++$;
        Flip $(1,3)$, $j++$;
    Case $m = 4$: For $i = 5$ To $n-2$
            { Flip $(1,i)$, $j++$; }
        Flip $(n-2,n)$, $j++$;
    Case $m = 5, 6, \cdots, n-3$:
        For $i = m+1$ To n-2
            { Flip $(1,i)$, $j++$; }
        Flip $(n-2,n)$, $j++$;
        For $i = m-2$ To 3

```
                    { Flip (1,i), j++; }
        Case m=n-2: Flip (1,n-4), j++;
                    Flip (1,n-3), j++;
                    For i=n-5 To 3
                        { Flip (1,i), j++; }
        Case m=n-1: Flip (n-2,n), j++;
                    Flip (1,n-3), j++;
                    Flip (1,n-2), j++;
                    For i=n-4 To 3
                        { Flip (1,i), j++; }
        Case m=n: For i=n-2 To 3
                        { Flip (1,i), j++; }
                    Flip (n-2,n), j++;
Step 6. Output j;
```

In Step 1, labelling T_0, and we get $Dia\_deg(1) = n-4$, $Dia\_deg(2) = 0$, $Dia\_deg(n-2) = 2$, $Dia\_deg(n-1) = 0$, and the other vertices are all with diagonal degree 1. In Step 5, we divide m ($\in \{2,3,\cdots,n\}$) into following cases

(1) if $m = 2,3,n-1,n$, from Lemma 1, the flip distance is minimal: $Dist(T_0,T_f) = n-3$;

(2) if $m = 4$ or $n-2$, from Lemma 1 and Lemma 2, the flip distance is minimal: $Dist(T_0,T_f) = n-5$;

(3) if $m = 5,6,\cdots,n-3$, from Lemma 1, Lemma 2 and Lemma 3, the flip distance is minimal: $Dist(T_0,T_f) = n-4$;

Similar to Algorithm 1, we reach the result:

Theorem 2. *The time complexity of Algorithm 2 is $O(n)$, and the diagonal-flip distance in type-II triangulations is $n-5$, or $n-4$, or $n-3$.*

C. DFD Algorithm in Type-III Triangulations

For $T_0, T_f \in L_3$, if we change the counterclockwise for clockwise in the labeling process of Algorithm 2, we can get "Algorithm 3" — DFD algorithm in type-III triangulations, and gain that DFD in type-III triangulations is $n-5$, or $n-4$, or $n-3$. Similar to Algorithm 2, Algorithm 3 can be done in $O(n)$ time.

D. DFD Algorithm between Type-I and -II Triangulations

Algorithm 4. DFD between type-I and -II triangulations

```
Input: T0 ∈ L2, Tf ∈ L1
Output: Dist(T0,Tf)
Step 1. In T0, seek the maximum degree vertex
        and label n vertices;
Step 2. Label the n vertices of Tf;
Step 3. m= the label of the maximum degree
vertex in Tf;
Step 4. j=0;
Step 5. Flip diagonals from T0:
        Case m=1: Flip (n-2,n), j++;
        Case m=2: For i=3 To n-2
                        { Flip (1,i), j++; }
                    Flip (n-2,n), j++;
        Case m=3,4,···n-3:
                    For i=m+1 To n-2
                        { Flip (1,i), j++; }
                    Flip (1,n-2), j++;
                    Flip (n-2,n), j++;
                    If m>3
                        For i=m-1 To 3
                            { Flip (1,i), j++; }
        Case m=n-2: For i=n-3 To 3
                        { Flip (1,i), j++; }
        Case m=n-1: Flip (n-2,n), j++;
                    For i=n-2 To 3
                        { Flip (1,i), j++; }
        Case m=n: For i=n-2 To 3
                        { Flip (1,i), j++; }
Step 6. Output j;
```

Labelling T_0, we get $Dia\_deg(1) = n-4$, $Dia\_deg(2) = 0$, $Dia\_deg(n-2) = 2$, $Dia\_deg(n-1) = 0$, and the other vertices are all with diagonal degree 1. Step 5 divides m into following cases:

(1) if $m = 1$, from Lemma 1 and Lemma2, the flip distance is minimal: $Dist(T_0,T_f) = 1$;

(2) if $m = 2$ or $m = n-1$, from Lemma 2, the flip distance is minimal: $Dist(T_0,T_f) = n-3$;

(3) if $m = n-2$, from Lemma 1 and lemma 2, the flip distance is minimal: $Dist(T_0,T_f) = n-5$;

(4) if $m = n$ or $m = 3,4,\cdots,n-3$, from Lemma 1, Lemma 2 and Lemma 3, the flip distance is minimal: $Dist(T_0,T_f) = n-4$.

Similarly, we reach the result:

Theorem 3. *The time complexity of Algorithm 4 is $O(n)$, the diagonal-flip distance between type-I triangulations and type-II triangulations is 1, or $n-5$, or $n-4$, or $n-3$.*

E. DFD Algorithm between type-I and -III triangulations

If we change the counterclockwise for clockwise in the labelling process of algorithm 4, we can get "Algorithm 5" — DFD algorithm between type-I and type-III triangulations, and gain that DFD between type-I triangulations and type-III triangulations are 1, or $n-5$, or $n-4$, or $n-3$. Similar to Algorithm 4, Algorithm 5 can be done in $O(n)$ time.

F. DFD algorithm between type-II and -III triangulations

Algorithm 6. DFD between type-II and -III triangulations

```
Input: T0 ∈ L2, Tf ∈ L3
Output: Dist(T0,Tf)
Step 1. In T0, seek the maximum degree vertex
        and label n vertices;
Step 2. Label the n vertices of Tf;
Step 3. m= the label of the maximum degree
vertex in Tf;
Step 4. j=0;
Step 5. Flip diagonals from T0:
        Case m=1: Flip (n-2,n), j++;
                    Flip (1,3), j++;
        Case m=2: Flip (1,4), j++;
                    Flip (1,3), j++;
                    For i=5 To n-2
                        { Flip (1,i), j++; }
                    Flip (n-2,n), j++;
        Case m=3,4,···,n-5:
                    Flip (1,m+2), j++;
                    Flip (1,m+1), j++;
                    For i=m+3 To n-2
                        { Flip (1,i), j++; }
                    Flip (n-2,n), j++;
                    If m>3
                        For i=m-1 To 3
                            { Flip (1,i), j++; }
```

```
        Case m = n-4: For i=m-1 To 3
                          { Flip (1,i), j++; }
                      Flip (1,n-3), j++;
                      Flip (1,n-2), j++;
                      Flip (n-2,n), j++;
                      Flip dual((1,n-3)), j++;
        Case m = n-3: Flip (1,n-2), j++;
                      For i = n-4 To 3
                          { Flip (1,i), j++; }
        Case m = n-2: Flip (n-2,n), j++;
                      For i = n-3 To 3
                          { Flip (1,i), j++; }
        Case m = n-1: Flip (n-2,n), j++;
                      For i = n-2 To 3
                          { Flip (1,i), j++; }
                      Flip dual((n-2,n)), j++;
        Case m = n: For i = n-2 To 4
                          { Flip (1,i), j++; }
Step 6. Output j;
```

Labelling of T_0, we know $Dia\_deg(1) = n-4$, $Dia\_deg(n-2) = 2$, $Dia\_deg(2) = 0$, $Dia\_deg(n-1) = 0$, and the other vertices are al with diagonal degree 1. Step 5 divides m into following cases:

(1) if $m = 1$, from Lemma 1 and Lemma 2, we know the flip distance is minimal: $Dist(T_0, T_f) = 2$;

(2) if $m = 2$, from Lemma 2, we know the flip distance is minimal: $Dist(T_0, T_f) = n-3$;

(3) if $m = n$ or $m = n-3$, from Lemma 1 and Lemma 2, we know the flip distance is minimal: $Dist(T_0, T_f) = n-5$;

(4) if $m = n-2$, from Lemma 1 and Lemma 2, we know the flip distance is minimal,: $Dist(T_0, T_f) = n-4$;

(5) if $m = 3, 4, \cdots, n-5$, from Lemma 1, Lemma 2 and Lemma 3, we know the flip distance is minimal: $Dist(T_0, T_f) = n-4$;

(6) for $m = n-4$, firstly $(1, n-4)$ must not be flipped following Lemma 1; secondly, $(1, n-5), (1, n-6), \cdots, (1, 3)$ are flipped following Lemma 2. Since the three diagonals $(1, n-3), (1, n-2), (n-2,)$ are not in T_f, so the three diagonals must be flipped.

How to flip the three diagonals $(1, n-3), (1, n-2), (n-2,)$? Suppose that first flip $(1, n-3)$, then need to flip $(1, n-2), (n-2,)$ in sequence according to Lemma 2, finally flip $dual((1, n-3))$; Suppose first flip $(1, n-2)$, then need to flip $(1, n-3), (n-2,)$ in sequence according to Lemma 2, finally flip $dual((1, n-2))$; Suppose first flip $(n-2,)$, then need to flip $(1, n-2), (1, n-3)$ in sequence according to Lemma 2, finally flip $dual((1, n-2))$.

From the above discussion, we know for $m = n-4$, the three flip methods are all minimal flips and $Dist(T_0, T_f) = n-3$.

(7) for $m = n-1$, first we know $Dist(T_0, T_f) \geq n-3$, then similar to the discussion of $m = n-4$, we know $Dist(T_0, T_f) \neq n-3$. So for $m = n-1$, the flip distance in the algorithm is minimal: $n-2$.

Similarly, we reach the result:

Theorem 4. *The time complexity of Algorithm 6 is $O(n)$, the diagonal-flip distance between type-II triangulations and type-III triangulations is 2, or $n-5$, or $n-4$, or $n-3$, or $n-2$.*

IV. The Rotation Distance between Binary Trees

By the well-known explicit bijection between binary trees and triangulations, we can conclude the rotation distance between binary trees with $n-2$ internal nodes:

Corollary

(1) *The rotation distance between binary trees according to type-I triangulations is $n-4$, or $n-3$.*

(2) *The rotation distance between binary trees according to type-II triangulations is $n-5$, or $n-4$, or $n-3$.*

(3) *The rotation distance between binary trees according to type-III triangulations is $n-5$, or $n-4$, or $n-3$.*

(4) *The rotation distance between binary trees according to type-I and type-II is 1, or $n-5$, or $n-4$, or $n-3$.*

(5) *The rotation distance between binary trees according to type-I and type-III is 1, or $n-5$, or $n-4$, or $n-3$.*

(6) *The rotation distance between binary trees according to type-II and type-III is 2, or $n-5$, or $n-4$, or $n-3$, or $n-2$.*

V. Conclusions

There is an open problem whether diagonal-flip (rotation) distance can be computed in polynomial time. However, in this paper, we have considered special triangulations, and by labelling vertices of the triangulations, we have given several linear time algorithms to compute the exact diagonal-flip distance between them. As a by-product, we have concluded the rotation distance between binary trees corresponding to the special triangulations.

References

[1] J. Baril, J. Pallo, "Efficient lower and upper bounds of the diagonal-flip distance between triangulations", *Inform. Process. Lett.*, 2006, 131-136.

[2] Y.J. Chen, J.M. Chang, Y.L. Wang, "An efficient algorithm for estimating rotation distance between two binary trees", *Intern. J. Comput. Math.*, 2005, 1095-1106.

[3] S. Cleary, J. "Taback, Bounding restricted rotation distance", *Inform. Process. Lett.*, 2003, 251-256.

[4] K. Culik, D. Wood, "A note on some tree similarity measures", *Inform. Process. Lett.*, 1982, 39-42.

[5] S. Hanke, T. Ottmann, S. Schuierer, "The edge-flipping distance of triangulations", *J. Universal Computer Science*, 1996, 570-579.

[6] F. Hurtado, M. Noy, J. Urrutia, "Flipping edges in triangulations", *Discrete Comput. Geom.*, 1999, 333-346.

[7] J.M. Lucas, "A direct algorithm for restricted otation distance", *Inform. Process. Lett.*, 2004, 129-134.

[8] J.M. Lucas, "Untangling binary trees via rotations", *Comput. J.*, 2004, 259-269.

[9] J.M. Pallo, "Right-arm rotation distance between binary trees", *Inform. Process. Lett.*, 2003, 173-177.

[10] J.M. Pallo, "An efficient upper bound of the rotation distance of binary trees", *Inform. Process. Lett.*, 2000, 87-92.

[11] D.D. Sleator, R.E. Tarjan, W. Thurston, "Rotation distance, triangulations and hyperbolic geometry", *J. Amer. Math. Soc.*, 1988, 647-681.

[12] R.-Y. Wu, "A linear time algorithm for binary tree sequences transformation using left-arm and right-arm rotations", *Theoretical Computer Science*, 2006, 303-314.