アルゴリズム 55−7
（1997. 1. 23）

警備員経路アルゴリズムの時間解析について

譚 学厚　　　平田 富夫
東海大学 開発工学部　　名古屋大学 工学部

警備員経路問題とは、与えられた多角形Ｐに対し、Ｐの任意の点が経路上の少なくとも１点から見えるような最短の経路を見つけることである。この問題を解くいくつかのアルゴリズムが既に提案されている。しかし、これらのアルゴリズムの時間解析には誤りがあることが最近指摘された。それを直すため、本論文ではこれまでの時間解析を修正し、既存のアルゴリズムの正当性を示す。

On the time analysis of watchman route algorithms

Xuehou TAN
School of High-Technology for Human Welfare, Tokai University
317 Nishino, Numazu 410-03, Japan
Tomio HIRATA
School of Engineering, Nagoya University
Chikusa-ku, Nagoya 464, Japan

Abstract

The watchman route problem is to find a shortest route in a simple polygon such that each point in the interior of the polygon is visible from at least one point along the route. The best time bound for this problem among three existing watchman route algorithms is $O(n^2)$, where n is the number of vertices of the given polygon. However, an error in the analysis of time complexities of those algorithms was recently pointed out. In this paper, we revise the method for analyzing the time complexities of the existing watchman route algorithms, and show that their time bounds are still correct.

1 Introduction

The *watchman route problem* [1, 2, 8, 9], an interesting variation of the well-known art gallery problem, deals with finding a route (actually a cycle) in a simple polygon with n edges so that each point in the interior of the polygon can be seen from at least one point along the route. Two points inside a polygon are said to be mutually visible if the segment between them lies entirely in the polygon. The objective is to minimize the length of the route. The watchman route problem deals with visibility as well as metric information.

We consider the watchman route problem for simple polygons that have a starting point s specified on their boundaries, i.e., watchman routes start at s and also end at s. In 1991, Chin and Ntafos [2] presented an $O(n^4)$ time algorithm for constructing shortest watchman routes in simple polygons. Their algorithm first finds an initial watchman route and then computes the shortest watchman route by repeatedly adjusting the current one. An adjustment always produces a new and shorter watchman route. Tan et al. improved the time bound to $O(n^3)$ [8] and later to $O(n^2)$ [9]. It is shown in [8] that a *one-place-adjustable* watchman route can be quickly adjusted into the shortest one.

However, an error in the analysis of time complexities of those algorithms was recently pointed out by Hammar [6]. In this paper, we revise the method for analyzing the time complexities of the existing watchman route algorithms, and show that their time bounds are still correct.

2 Definitions

Let P be an n-sided simple polygon, given by the sequence of its vertices in the clockwise order from s. A vertex is *reflex* if the internal angle is strictly greater than 180^0. P can be partitioned into two pieces by a "cut" that starts at a reflex vertex v and extends either edge incident to v until it first intersects the boundary of P. We say a cut is a *visibility cut* if it produces a *convex* angle ($< 180^0$) at v in the piece of P containing s. In order to see the corner incident to a visibility cut, a watchman route needs to visit at least one point on that cut. The piece of P containing s is called the *essential piece* of C. All visibility cuts can be found in $O(n \log n)$ time using the ray-shooting algorithm [3].

We say cut C_j *dominates* cut C_i if the essential piece of C_j contains that of C_i. Clearly, if C_j dominates C_i, any route that visits C_j will automatically visit C_i. A cut is called an *essential cut* if it is not dominated by any other cuts. The watchman route problem is now reduced to that of finding the shortest route that visits all essential cuts. In the rest of this paper, we consider only the essential cuts. Let m be the number of essential cuts and let $C_1, C_2, \cdots, C_m$ be the sequence of essential cuts indexed in the clockwise order of their left endpoints. A cut can intersect with some cuts and is thus divided into several segments. We call these segments the *fragments* of a cut. We say fragment f (point p) *dominates* cut C if f (p) lies in the non-essential piece of C. That is, any route that visits f (p) also visits the cut C. We also say fragment f *dominates* fragment g if f dominates the cut to which g belongs.

Lemma 1 *(Chin and Ntafos [2]) The shortest watchman route should visit the cuts in the order in which they appear in the boundary of P.*

Depending on whether a watchman route goes over a cut (as viewed from s) or just reflects on a cut, we say that the watchman route makes a *crossing contact* or a *reflection contact* with the cut, respectively. The degenerate cases of reflection contacts and crossing contacts, where the watchman route shares a line segment with the cut, are called *tangential contacts* (see also [2, 8] for exact definitions). For a watchman route, the set of the fragments with which the route makes reflection contacts is called the *watchman fragment set.* With respect to a watchman fragment set, we distinguish a fragment as an *active* or *inactive* fragment according to whether it belongs to the fragment set or not. A cut is *active* if it contains an active fragment. Otherwise, it is *inactive.*

Given a watchman fragment set, we can construct the corresponding (optimal) watchman route in polygon P. Specifically, the non-essential pieces of all active cuts are removed (since the optimal watchman route never needs to enter them) and the resulting polygon P' is then triangulated. The active fragments are used in order as mirrors to "roll-out" the triangulation of P'. Now, the optimal watchman route can be determined by finding the shortest route between s and the image t of s

in the rolled-out polygon, and by folding back the shortest route along the active fragments. The whole procedure takes $O(n)$ time since a simple polygon can be triangulated in linear time [4] and the shortest path between two points in a triangulated polygon can be found in linear time [5]. (See [2] for details.) We call this method as the rolled-out method. Note that the watchman route found by the rolled-out method is optimal only with respect to the given watchman fragment set. It is thus possible to make it shorter by changing the watchman fragment set.

Definition 1 *A watchman route R is adjustable on an active cut C if (i) R makes a reflection contact with C, (ii) the incoming angle of R with C is not equal to the outgoing angle and (iii) the contact point of R with C is not the endpoint of C. (That is, we can adjust the contact point on C to get a shorter watchman route.)*

Definition 2 *A watchman route R is one-place-adjustable if R is adjustable only on one active cut.*

As shown in [2], there are three types of adjustments on an active cut C_i. In Fig. 1, the incoming angle of the current route R^l with C_i is assumed to be smaller than the outgoing angle. (The symmetric case is omitted.) Thus, we should adjust the reflection contact of C_i to the left. The bold and discontinuous segments in Fig. 1 stand for the active fragments before and after an adjustment, respectively. A possible next route R^{l+1} is also shown. An adjustment occurs only at the intersection of two essential cuts and causes some changes in the watchman fragment set. For such a change, we have to compute the new watchman route by the rolled-out method. Depending on the change of the number of active fragments, we call them as *(-1)-adjustments* (Fig. 1a), *0-adjustments* (Fig. 1b), *(+1)-adjustments* (Fig. 1c) and *(+1)-switches* (Fig. 1d), respectively. Note that a *(+1)-switch* occurs only when the incoming angles of R^l with two considered cuts are smaller than the outgoing angles. In this case, $R^l = R^{l+1}$ and furthermore it follows a (-1)-adjustment in R^{l+1} that makes C_i inactive.

Theorem 1 *(Chin and Ntafos [2]) There is a unique non-adjustable watchman route in a simple polygon P.*

Lemma 2 *(Chin and Ntafos [2]) A watchman route R is a shortest watchman route if and only if R is non-adjustable.*

3 Time analysis of watchman route algorithms

We first define the direction of adjustments on an active cut.

Definition 3 *The direction of an adjustment on an adjustable cut C is the shift direction of the reflection points on C.*

Chin and Ntafos' algorithm first finds an initial watchman route and then computes the shortest watchman route by repeatedly adjusting the current one with the *adjust at the first choice* selection rule. That is, the adjustment on the cut of the smallest index is done first. In the proofs of Lemma 8 and Theorem 2 of [2], Chin and Ntafos made an observation that the adjustments resulting from an adjustment on C_j should be all in the same direction on the previous cuts. Tan et al. dealt with an initial *one-place-adjustable* watchman route and made the similar argument in the time analysis of their algorithm (Lemmas 5 and 6 of [8]). However, Hammar recently pointed out that Chin and Ntafos' observation was not generally true and the adjustments on the cuts of lower indexes would change their directions because of a $(+1)$-adjustment [6]. (Actually, such phenomenon also occurs for a (-1)-adjustment.)

In the following, we show that the mistake found by Hammar is not so serious. Let R^0 denote a one-place-adjustable watchman route. We will show that in the process of computing the shortest watchman route from R^0, although a 0-adjustment at the same place (cut intersection) may be repeatedly performed several times, neither $(+1)$-adjustments nor (-1)-adjustments can be repeated. Hence, at most $O(m)$ $(+1)$- and (-1)-adjustments and $O(m^2)$ 0-adjustments can be made. Note that $(+1)$-adjustments and (-1)-adjustments are time-consumed; each of them requires to redraw the rolled-out polygon and thus spends linear time, but a 0-adjustment can be implemented in constant time by the new technique introduced in this paper. Thus, the above mistake can be corrected by changing the method for analyzing the time complexities of the existing watchman route algorithms.

Assume that C_i is the only adjustable cut of route R^0 and r_i is the reflection point on C_i. To find the shortest watchman route from R^0, we first generalize the *adjust at the first choice* rule to what we call the *adjust at the first/last choice* rule. That is, an adjustment on C_i is made only when the current route is one-place-adjustable, and the adjustments on two parts, one from s to r_i and the other from r_i to t, are performed using the *adjust at the first choice* and *adjust at the last choice* selection rules, respectively. Thus, the reflection point on C_i can not oscillate. The first and last adjustments (not on C_i) can be performed simultaneously, as the adjusting effect is isolated by C_i.

Without loss of generality, we assume that the adjustable cut C_i remains active in the whole adjusting process. (If C_i becomes inactive because of a (-1)-adjustment, then the act of C_i is replaced by the other active cut. Thus, this assumption can always be satisfied.) To catch characteristics of $(+1)$- and (-1)-adjustments, we slightly deform the rolled-out method as follows. Instead of starting from s in the original method, we begin with cut C_i, and roll-out the portion of the watchman route from r_i to s to the left side of C_i and the portion from r_i to t to the right side of C_i, respectively. In this way, the position of C_i is always fixed, but the positions of both s and t are often changed in the rolled-out polygons. Since R^0 is one-place-adjustable, the rolled-out version of R^0 can be topologically considered as two line segments $\overline{sr_i}$ and $\overline{r_it}$ (in the sense neither of them can further be shortened), and is thus a convex chain. Note that the line segment $\overline{st}$ can be considered as the target that the current watchman route tries to reach. In the following, we show that the rolled-out version of the route R^l $(l \geq 0)$ always appears as a convex polygonal chain in the adjusting process, and the change of positions of both s and t in the rolled-out polygons is opposite to the direction in which r_i moves.

Since a 0-adjustment does not change the set of active cuts, the rolled-out versions of routes R^l and R^{l+1} $(l \geq 0)$ before and after the adjustment are superimposed with R^{l+1} interior to R^l. Thus for a 0-adjustment, the positions of s and t are not changed and the convexity of the rolled-out routes is simply kept.

Consider now $(+1)$- and (-1)-adjustments. Assume first that a $(+1)$-adjustment (Fig. 1c) on C_i is chosen to be done. Note that the current route R^l is one-place-adjustable in this case. Let C_j denote the other participating cut. If $i < j$, then the incoming angle of R^l with cut C_j has to be greater than the outgoing angle (see also Fig. 1c). So C_j is added as a new mirror when we use the rolled-out method to compute R^{l+1}. It results in a reflection of the segment $\overline{r_it}$ along C_j in the rolled-out polygon. For comparison, two routes with the same length, immediately before and after the $(+1)$-adjustment, are shown in Fig. 2. Since the incoming angle of R^l with C_j is greater than the outgoing angle, both the line segment $\overline{r_it}$ and its reflection image lie in one side of the line extending $\overline{sr_i}$. Clearly, t moves to a new position in the direction opposite to which r_i moves, and the rolled-out route R^{l+1} remains as a convex polygonal chain (Fig. 2). Similarly if $i > j$, then the incoming angle of R^l with cut C_j has to be smaller than the outgoing angle, and the line segment

$\overline{sr_i}$ is reflected along C_j in the rolled-out polygon.

Let us consider now the (+1)-switch and the (−1)-adjustment occurred on C_i. Note that a (+1)-switch should be followed by a (−1)-adjustment, and that C_i can take part in a (−1)-adjustment only after a (+1)-switch on C_i is performed and this (−1)-adjustment makes C_i inactive (see also Fig. 1d). Let C_j denote the cut participating the (+1)-switch and the (−1)-adjustment. In this case, the incoming angles of R^l with C_i and C_j are smaller than outgoing angles. It acts to switch the only adjustable cut of the current route from C_i to C_j. Switching from C_i to C_j asks s to jump to a new position opposite to the direction in which r_i moves in the rolled-out polygon. See Fig. 3. For convenience, the image of s before reflecting the part from s to r_i along C_i or C_j, denoted by s', is also shown in Fig. 3. Again, the convexity of the new rolled-out route is kept.

If the adjustment to be made occurs on the cut C_h ($h \neq i$), we assume without loss of generality that $h < i$ and that C_h is the only adjustable cut in the part of the current route from s to r_i. Let C_k denote the other cut taking part in the (+1)- or (−1)-adjustment on C_h. If it is a (+1)-adjustment, then C_h is active in both R^l and R^{l+1} (Fig. 4a and Fig. 4b), and two active fragments of C_k and C_h have to be interior to the convex chain of the route shown in Fig. 4b. Since r_i is closer to C_h than C_k in the rolled-out figure when both C_k and C_h are active, index k is smaller than index h. Thus, the line segment $\overline{sr_h}$ is reflected along C_k to account for the (+1)-adjustment. As the incoming angle of R^l with C_k (C_h) is smaller (greater) than the outgoing angle in this case, $\overline{sr_h}$ and its reflection image lie in one side of the line extending $\overline{r_h r_i}$. Hence, the convexity of the rolled-out route R^{l+1} is kept, and s jumps to a position opposite to the direction in which r moves (Fig. 4). Similarly, for a (−1)-adjustment, C_h is active in both R^l and R^{l+1} (Fig. 5a and Fig. 5d), and two active fragments of C_k and C_h have to be outside of the convex chain of the route shown in Fig. 5a. In this case, index k is smaller than index h, and the incoming angle of the route with C_h is smaller that the outgoing angle (Fig. 5a-b), even after C_k becomes inactive (Fig. 5c-d). Again, $\overline{sr_h}$ is reflected to account for the (−1)-adjustment, and the convexity of the current rolled-out route is kept. In order to show the new position of s after the (−1)-adjustment, two intermediate steps are shown in Fig. 5b-c. (As a result, the position of segment $\overline{sr_h}$ in Fig. 5d can be simply obtained by reflecting segment $\overline{sr_h}$ along C_k in Fig. 5a.)

Since C_h is the first adjustable cut in route R^l and the rolled-out version of R^l is a convex chain, the discussion above also works if the assumption that C_h is the only adjustable cut in the part from s to r_i is removed. In summary, we have the following result.

Lemma 3 *Let R^0 be a one-place-adjustable route and let C_i be the only adjustable cut of R^0. Every (+1)- or (−1)-adjustment chosen by the* **adjust at the first/last choice** *rule asks s or t to move in the reverse direction of adjustments on C_i, and the convexity of the rolled-out routes is kept in the whole adjusting process.*

By now we can give our main result.

Theorem 2 *The shortest watchman route can be found from a one-place-adjustable route in $O(n^2)$ time using the* **adjust at the first/last choice** *selection rule.*

Proof. Let R^0 denote the one-place-adjustable route, and let r_i be the reflection point on the adjustable cut C_i of R^0. For simplicity, assume that the *adjust at the first/last choice* rule first selects the adjustments in the part of R^l from s to r_i and then in the part from t to r_i. When there exist no adjustments in both parts, the adjustment on C_i, if exists, is made.

First, let us consider the number of (+1)- and (−1)-adjustments required in the process of computing the shortest watchman route from R^0. We claim that once an active cut becomes inactive because of a (-1)-adjustment in the adjusting process, it can never be active again. If the (-1) adjustment occurs on C_i, then as discussed above, C_i has to become inactive. Since the route after the (-1)-adjustment is adjustable only on the other participating cut (e.g., C_j in Fig. 3), C_i can not be active again. Assume now that the (-1) adjustment on cut C_h makes cut C_k inactive and that $h, k < i$. See Fig. 5 for an example. From Lemma 3, the active fragment of C_h will dominate C_k since then (Fig. 5d). If C_h becomes inactive because of a (-1)-adjustment on a cut $C_{h'}$, then $C_{h'}$ will dominate C_h and C_k. By a simple induction proof, we can conclude that C_k can never be active again. Thus, the claim is proved. It then follows that the total number of $(+1)$ and (-1)-adjustments in the whole process is $O(m)$, which takes $O(mn)$ time. (The same observation is also made in Lemma 6 of [8].)

Let us now analyze the time required by 0-adjustments. Between two of (+1)- and/or (-1)-adjustments, the set of active cuts is not changed. Since the current rolled-out route R^i is a convex chain and the directions of 0-adjustments all point to the line segment $\overline{st}$, the direction of 0-adjustments on any active cut can never be changed between two of (+1)- and/or (-1)-adjustments, no matter which 0-adjustment is first performed (otherwise the length of the current route can not be shortened). So the number of 0-adjustments between two of (+1)- and/or (-1)-adjustments is bounded by number of fragments on active cuts. As active cuts forms several disjoint convex chains within the given polygon, an inactive cut can intersect with at most two active cuts, and thus contributes to at most two fragments on active cuts. So this number of fragments on active cuts is bounded by $O(m)$.

If we perform each 0-adjustment independently, then $O(m)$ 0-adjustments take $O(mn)$ time. However, since the set of active cuts is not changed for these 0-adjustments, we can group them together and adjust only once so that these adjustments can be made in linear time. First, since the current rolled-out route R^l is a convex chain and the directions of 0-adjustments all point to the line segment $\overline{st}$, the 0-adjustments on different active cuts can be grouped together. Such 0-adjustments can be found in linear time using the *adjust at the first/last choice* selection rule. The remaining problem is, if a 0-adjustment on an active cut C is found, then how many 0-adjustments (or fragments) on C can be safely grouped together. If the whole cut C, instead of the active fragment of C, is used in the rolled-out menthod, then the resulting route might not visit all of the cuts. Therefore, we need to give a divide on cut C so that the fragments between the current reflection point on C and the divide can be safely grouped together (i.e., the resulting route will reflect on C at some point of that interval).

As discussed above, the divide on C is determined by the next possible $(+1)$-adjustment or $(+1)$-switch on C. Let us consider the possibility for an inactive cut C' to take part in a $(+1)$-adjustment or $(+1)$-switch on C. It may occur when the reflection point of the current route on C is the only reflection point that lies in the non-essential piece of C' and moves towards C' (see also Fig. 1c-d). So we check the cut intersections in the adjusting direction on C to find the first inactive cut satisfying the above condition. If such a cut C' is found, then the intersection of C and C' is the divide of C. If no inactive cuts satisfy the above condition, then the whole cut C can be used as a mirror in the rolled-out method. It is easy to see that the divide on an active cut C can be found in time linear to the number of inactive cuts intersecting C, provided that all fragments and the order of the fragments on each cut are precomputed in $O(m^2)$ time. Thus, we can find the divides for all active cuts in $O(m)$ time. After the divides are found, we perform all of those (possible) 0-adjustments once, which takes only $O(n)$ time. Since there are at most $O(m)$ (+1)- and (-1)-adjustments in the whole adjusting process, the total time taken by 0-adjustments is $O(mn)$. It completes the proof. □

If the method shown in Theorem 2 is used in the time analysis of existing watchman route algorithms [2, 8, 9], then all of previous time bounds are correct. Thus, Tan and Hirata's divide-and-conquer algorithm [9] solves the watchman route problem in $O(n^2)$ time, which is the best bound of three existing watchman route algorithms.

4 Conclusion

We have revised the method for analyzing the time complexities of watchman route algorithms, and shown the correctness of time bounds of existing algorithms. Our analysis depends on the convexity of watchman routes in the adjusting process, which is shown by slightly deforming the rolled-out method. We also present a new technique for grouping a set of 0-adjustments as many as possible, which ensures the efficiency of previous watchman route algorithms.

Carlsson, Jonsson and Nilsson have presented an $O(n^4)$ time algorithm for computing an overall shortest watchman route without giving any start point, in which Tan and Hirata's $O(n^2)$ time algorithm is used as a subroutine. For detail, see [7].

References

[1] W.P.Chin and S.Ntafos, Optimum watchman routes, *Inform. Process. Lett.* **28**, 39-44, 1988.

[2] W.P.Chin and S.Ntafos, Shortest watchman routes in simple polygons, *Discrete Comput. Geometry* **6**, 9-31, 1991.

[3] B. Chazelle and L. J. Guibas, Visibility and intersection problems in plane geometry, *Discrete Comput. Geometry* **4**, 551-581, 1989.

[4] B. Chazelle, Triangulating a simple polygon in linear time, *Proceedings, 31th Annu. IEEE Symp. Found. of Comput. Sci.*, 220-229, 1990.

[5] L.Guibas, J.Hershberger, D.Leven, M.Sharir and R.Tarjan, Linear time algorithms for visibility and shortest path problems inside simple triangulated polygons, *Algorithmica* **2**, 209-233, 1987.

[6] M. Hammar, Concerning the time bounds of existing shortest watchman route algorithms, Master Thesis, Lund University, Sweden, 1996.

[7] B. Nilsson, Guarding art galleries; Methods for mobile guards, PhD thesis, Lund University, Sweden, 1995.

[8] X.Tan, T.Hirata and Y.Inagaki, An incremental algorithm for constructing shortest watchman routes, *Int. J. Comput. Geometry & Appl.* **3**, 351-365, 1993.

[9] X.Tan and T.Hirata, Constructing shortest watchman routes by divide-and-conquer, *LNCS* **762**, 68-77, 1993.

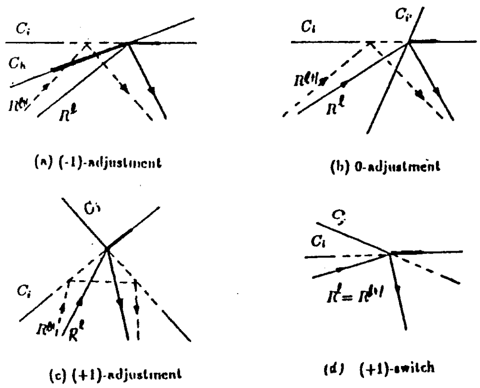

Fig. 1. Basic types of adjustments.

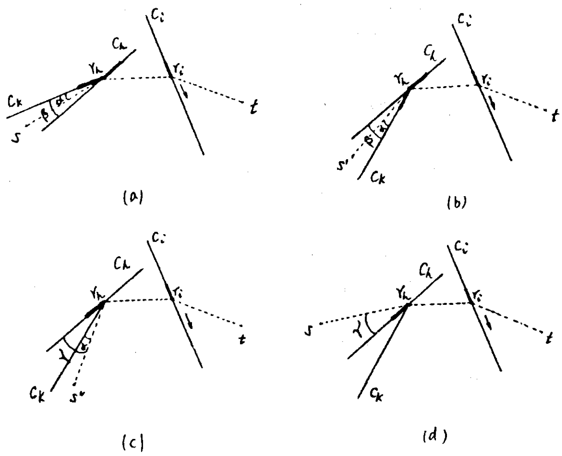

Fig. 5. A (-1)-adjustment on C_h with $h < i$.

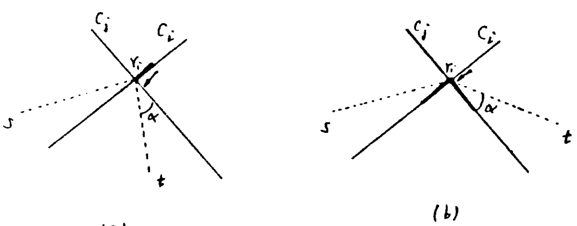

Fig. 2. Two routes with the same length before and after a $(+1)$-adjustment on C_i.

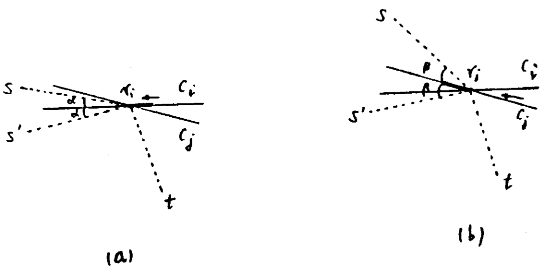

Fig. 3. A pair of a $(+1)$-swicth and a (-1)-adjustment on C_i.

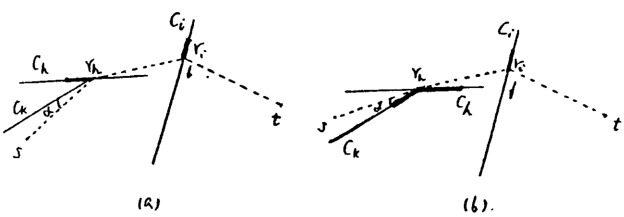

Fig. 4. A $(+1)$-adjustment on C_h with $h < i$.