Computing Tumour Coverage as a Result of Respiratory Motion During Radiotherapy Treatment

J. Chen
Carleton University

M. Lanthier
Carleton University

M. MacPherson
Ottawa Regional Cancer Centre

D. Nussbaum
Carleton University

Abstract

The objectives of radiotherapy treatment is to kill cancerous cells while minimizing damage to surrounding healthy tissues. The tumour location uncertainty "forces" oncologists to prescribe a larger treatment area than required in order to ensure that the whole tumour is receiving the prescribed dose. The problem is more acute when a tumour can move during treatment, e.g., as a result of breathing. In this paper, we present an algorithm for computing the area covered by a tumor as a result of a cyclic motion during treatment. Our algorithm solves the following geometric problem: Given an n*-vertices convex polygon* $P = \{v_1, v_2, \ldots, v_n\}$*, a monotone chain* $C = \{c_1, c_2, \ldots, c_m\}$*, compute a minimums area polygon* Q *that includes all the space covered by* P *as it is translated along* C *such that* $v_1 \in P$ *touches* C*. Here, we present a simple algorithm when* P *is a convex polygon. Our algorithm takes* $O(mn + m \log n \log(m + n))$ *time in the worst case.*

1 Introduction

The objective of radiation treatment is to kill cancerous cells while minimizing damage to surrounding healthy tissues. Although oncologists determine a Clinical Target Volume (CTV) they must prescribe a larger Planning Target Volume (PTV) by adding a safety margin around the CTV in order to ensure that the tumour is completely exposed to radiation during treatment. A number of factors contribute to the final size of the PTV, including setup errors, organ motion, physical and geometric umbra and penumbra. In the last decade, methods for reducing damage to healthy tissues have been studied and developed [4, 8, 9, 10, 11]. These methods include immobilizing devices and position tracking devices such as stereoscopic cameras and video based position tracking. Special consideration is given to tumour motion during treatment e.g., tumour motion as a result of the respiratory cycle [1, 6, 7, 12, 13, 14, 15]. Breathing cycle during treatment has a significant effect on the position of the internal anatomy.

The algorithm, which is presented in this paper, is a part of a larger system that is currently being developed at Carleton University together with the Ottawa Regional Cancer Centre (ORCC). The system focuses on developing means for tracking the tumour motion and then verifying the treatment and potentially redefining the treatment fields [2]. In this system a sequence of x-ray images is captured in a simulator or in real-time during treatment. The x-ray images are then analyzed and the tumour is identified in each of the electronic portal images. Then the algorithm computes the regions occupied by the tumour during the cyclical breathing pattern of the patient for treatment verification or adjustment.

Our algorithm solves the following geometric problem: Given an n-vertices convex polygon $P = \{v_1, v_2, \ldots, v_n\}$, a monotone chain $C = \{c_1, c_2, \ldots, c_m\}$, compute a minimum area polygon Q that contains all the space covered by P as it is translated along C such that $v_1 \in P$ touches C. Here, we present a simple algorithm when P is a convex polygon. Our algorithm takes $O(mn + m \log n \log(n + m))$ time in the worst case.

Similar problems arise in computer graphics animation and in robotics. In computer animation a trajectory is defined for a given object and the goal is to determine the intermediate locations between the start position and the end position (e.g., the motion of a hand)[5]. In robotics a very similar problem arises in robot path planning. Here the objectives is to find a path for a robot from a starting point s to destination point d such that the robot does not collide with obstacles along the way. Algorithms which are based on Minkowski sums are often used to compute the configuration space of the robot. Once computed, a search for a valid path through the configuration space between s and t is executed [3].

The paper is organized as follows. In Section 2 we present notation and definitions used throughout the paper followed by the properties of a path polygon which we give in Section 3. In Section 4 we describe our algorithm and in Section 6 we summarize our result and discuss future work.

2 Preliminaries

In this section we present notations and definitions that are used throughout the paper.

Let $P = \{v_1, v_1, \ldots, v_n\}$ be a convex polygon whose vertices are given in counter clockwise (CCW) order and let $C = \{c_1, c_2, \ldots, c_m\}$ be a monotone chain. For simplicity we assume that C is a strictly decreasing monotone chain in the y direction, namely, $y(c_i) > y(c_{i+1}), 1 \leq i < m$. We also assume that P is translated along C such that $v_1 \in P$ touches C throughout the motion of P.

We denote by v_t the topmost vertex of P (the vertex $v \in P$ with the highest y-coordinate) and by v_b the bottommost vertex of P. Vertices v_t and v_b divide P into two monotone chains: ***the left chain***, denoted as LC whose vertices, in CCW order, are $\{v_1, \ldots, v_b\}$, and ***the right chain***, denoted as RC whose vertices, in CCW order, are $\{v_b, \ldots, v_1\}$.

In our problem, polygon P is translated along chain C. We denote by P^i the copy of P when it is positioned at vertex $c_i \in C$ and the vertices of P^i are $\{v_1^i, v_2^i, \ldots, v_n^i\}$. The topmost vertex, bottommost vertex, left chain and right chain of P^i are denoted by v_t^i, v_b^i, LC^i and RC^i, respectively.

Vertices $u \in P^i$ and $v \in P^j$ are ***extreme vertices***, if P^i and $P^j, i > j$, lie in the same half plane defined by a line L through u and v. Vertices u and v are ***left extreme vertices*** if P^i and P^j lie on the right side of L. Similarly, vertices u and v are ***right extreme vertices*** if P^i and P^j lie on the left side of L. Note that the line L is not directed by u and v and thus "right" and "left" are not related to a direction of L. The segment $\overline{uv}$ is termed ***extreme edge***.

The solution to our problem is the minimum area polygon which includes all the regions touched by P as it is translated along chain C. We term the output polygon ***path polygon***. We denote by, $P^{i,k}$, the output path polygon, when P is translated along a sub-chain $\{c_i, c_{i+1}, \ldots, c_k\}$, of C,. The topmost and bottommost vertices of $P^{i,k}$ are denoted as $v_t^{i,k}$ and $v_b^{i,k}$, respectively. Similarly, the left chain and the right chain of $P^{i,k}$ are denoted by $LC^{i,k}$ and $RC^{i,k}$, respectively.

3 Properties of Path Polygon

In this section we present several properties of the path polygons upon which our algorithm is based.

In our problem P is a convex polygon and, for simplicity, C is a strictly decreasing monotonic chain in the y-direction. Thus, we observe the following:

Observation 1 *Given a convex polygon P and a monotonically decreasing chain C in the y-direction, the path polygon, $P^{i,i+1}$, which is a result of translating P along segment $\overline{c_i c_{i+1}}$ of C is the convex hull of P^i and P^{i+1}.*

Proof:
If $P^{i,i+1} \neq Q$, where Q denote the convex hull of P^i and P^{i+1} ($Q = CH(P^i, P^{i+1})$), then two cases arise:

Case 1 $P^{i,i+1} \cap Q \neq Q$**:** in this case there is a point $o \in Q - (P^{i,i+1} \cap Q)$ and therefore $o \notin P^{i,i+1}$. Let L be a line parallel to $\overline{c_i c_{i+1}}$ through o and let v be the intersection between L and P^i. Let v' be a point on P^{i+1} such that when P^i is overlayed on P^{i+1} then $v = v'$. In this case $P^{i,i+1}$ does not contain all the segment $\overline{vv'}$ and therefore it cannot be a path polygon.

Case 2 $P^{i,i+1} \cap Q \neq P^{i,i+1}$**:** in this case we assume that Case 1 does not hold and therefore $P^{i,i+1} \cap Q = Q$. Thus, there exists a point $o \in P^{i,i+1} - (P^{i,i+1} \cap Q)$. Note, that $o \notin P^i$ and $o \notin P^{i+1}$ since $o \notin Q$. Let L be a line parallel to $\overline{c_i c_{i+1}}$ through o and let v be the intersection of L and P^i. Let v' be a point on P^{i+1} such that when P^i is overlayed on P^{i+1} then $v = v'$. Since $o \notin Q$ implies that line segment $\overline{vv'} \notin Q$ contradicting the fact that $Q = CH(P^i, P^{i+1})$.

□

The fact that $P^{i,i+1}$ is convex leads to the following corollary regarding the number of vertices in $P^{i,i+1}$.

Corollary 3.1 *Given a convex polygon $P = \{v_1, \ldots, v_n\}$ and a monotonically decreasing chain C in the y-direction, the number of vertices of the path polygon, $P^{i,i+1}$, which is a result of translating P along segment $\overline{c_i c_{i+1}}$ of C is $n+2$.*

The convexity of the path polygon does not hold when C consists of 3 or more vertices (unless C is a straight line). In this case the path polygon is monotone (see Lemma 3.1 below). Moreover, the top and bottom vertices of can be easily computed (see Lemma 2).

Observation 2 *Given a convex polygon P and a monotonically decreasing chain $C = \{c_i, c_{i+1}, \ldots, c_k\}$ in the y-direction, the topmost and bottommost vertices of the path polygon, $P^{i,i+k}$ are v_t^i and v_b^{i+k} respectively ($v_t^{i,i+k} = v_t^i$ and $v_b^{i,i+k} = v_b^{i+k}$).*

Proof: Omitted. □

Following the two observations, we proceed to show that the path polygon is monotone in the y-direction.

Lemma 3.1 *Given a convex polygon P and a monotonically decreasing chain C in the y-direction, the path polygon $P^{i,i+k}$ is* y *monotone.*

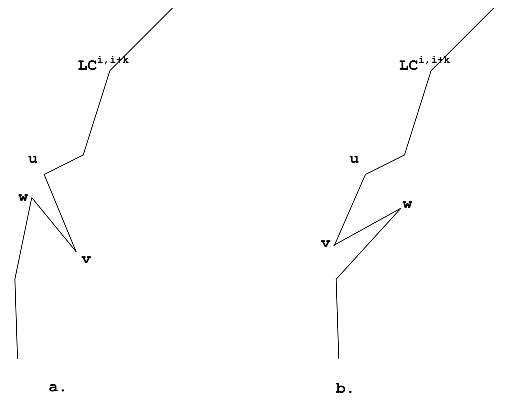

Figure 1. The two cases that the monotonicity of the $LC^{i,i+k}$ can be violated.

Proof:

By Lemma 2, $v_t^{i,i+k}$ and $v_b^{i,i+k}$ divide $P^{i,i+k}$ into two chains the $LC^{i,i+k}$ and $RC^{i,i+k}$. If $P^{i,i+k}$ is not y-monotone then either $LC^{i,i+k}$ or $RC^{i,i+k}$ are not y-monotone. Without loss of generality assume that $LC^{i,i+k}$ is not y-monotone. Then there are three consecutive vertices $u, v, w \in LC^{i,i+k}$ where the monotonicity of $LC^{i,i+k}$ is violated for the first time. Namely, the segment $\overline{vw}$ is added to such that $y(u) > y(v)$ and $y(w) > y(v)$, (see Figure 1). Assume that vertex v is a result of adding P^j to $P^{i,j-1}$ and vertex w is a result of adding P^{j+1} to $P^{i,j}$. There are two cases to examine:

Case 1 w **is to the right of** $\overline{uv}$ **(Figure 1b):** in this case P^j and P^{j+1} are to the left of $\overline{vw}$. Since $\overline{vw}$ has a positive slope it implies that $y(v_b^{j+1}) > y(v_b^j)$. This contradicts the monotonicity of C.

Case 2 w **is to the left of** $\overline{uv}$ **(Figure 1a):** in this case $w = v_t^{j+1}$ and from Lemma 2 $v = v_b^j$. Since C is a monotonically decreasing chain it implies that $y(v_t^{j+1}) < y(v_t^j)$ and that $y(v_b^{j+1}) < y(v_b^j)$. Polygon P^j is to the right of $\overline{uv}$ and therefore, $\overline{vw} \in CH(P^j, P^{j+1})$. This contradicts the assumption that $\overline{vw} \in LC^{i,j+1}$.

□

4 Algorithm

In this section we describe our algorithms for finding the path polygon of a convex polygon P: Given an n-vertices convex polygon $P = \{v_1, v_2, \ldots, v_n\}$, a monotonically decreasing chain $C = \{c_1, c_2, \ldots, c_m\}$, compute a minimum area polygon Q that contains all the regions covered by P as P is translated along C such that $v_1 \in P$ touches C.

Our algorithm is based on the facts that $P^{i-1,i}$ is convex (see Lemma 1) and that $P^{i,i+k}$ is monotone (see Lemma 3.1). The idea behind the algorithm is as follows. Starting from c_1, make a copy of P and place it at c_i to form P^i. Then add P^i to $P^{1,i-1}$ one at a time.

We first describe, in Section 4.1, how to add P^i to $P^{1,i-1}$, which is the crux of the algorithm. Then in Section 4.2 we present our algorithm and the time complexity.

4.1 Adding P^i to $P^{1,i-1}$

In this section we show how polygon P^i is added to $P^{1,i-1}$. Note that we only present the algorithmic steps of updating the left chain of $P^{1,i-1}$. Similar and symmetrical steps are taken to update the right chain of $P^{1,i-1}$.

Observation 3 *Let* $\overline{uv}$ *be the left extreme edge of* $CH(P^{i-1}, P^i)$ *where* $u \in P^{i-1}$ *and* $v \in P^i$. *When updating* $LC^{1,i-1}$ *there are two cases to consider.*

Case 1 *Vertex* $u \in LC^{1,i-1}$ *(see Figure 2a). In this case vertices* $u, v, \ldots, v_b^i$ *replace vertices* $u, \ldots, v_b^{1,i-1} \in LC^{1,i-1}$.

Case 2 *Vertex* $u \subset P^{1,i-1}$ *(*$u \notin LC^{1,i-1}$*) (see Figure 2b). In this vertices* $w, \ldots, v_b^{i-1,i}$ *are used to update* $LC^{1,i-1}$, *where* w *is the intersection between* $LC^{1,i-1}$ *and* $LC^{i-1,i}$.

Proof: In Case 1 $u \in LC^{1,i-1}$ and therefore the left extreme edge $\overline{uv}$ is outside $LC^{1,i-1}$. Thus, it must be added to $LC^{1,i-1}$. The remaining vertices, $v, \ldots, v_b^i$, are added by Lemma 1.

In Case 2, $u \notin LC^{1,i-1}$ ($u \subset P^{1,i-1}$) and therefore it cannot be part of $LC^{1,i}$. Since v_b^i must be a part of $P^{1,i}$ it implies that $LC^{i-1,i}$ and $LC^{1,i-1}$ must intersect. Once the intersection of $LC^{i-1,i}$ and $LC^{1,i-1}$ is found, the intersection point is added to $LC^{1,i-1}$ followed by vertices of $w, \ldots, v_b^i \in LC^{i-1,i}$ as a result of Lemma 1. □

Function AddPolygon($P^{1,i-1}$, P^i)
{The function adds polygon P^i to path polygon $P^{1,i-1}$ }
{Updating the left chain of $P^{1,i-1}$}
find the convex hull of P^{i-1}, P^i
$u \leftarrow$ left extreme vertex of P^{i-1}
$v \leftarrow$ left extreme vertex of P^i
if $u \in P^{1,i-1}$ **then** {see Figure 2a}
  remove vertices $u, \ldots, v_b^{1,i-1}$ from $LC^{1,i-1}$
  $LC^{1,i} \leftarrow LC^{1,i-1} \cup u, v, \ldots, v_b^i$
else {here $u \notin P^{1,i-1}$, thus, $LC^{i-1,i} \cap LC^{1,i-1}$ may intersect at a new vertex (see Figure 2b)}

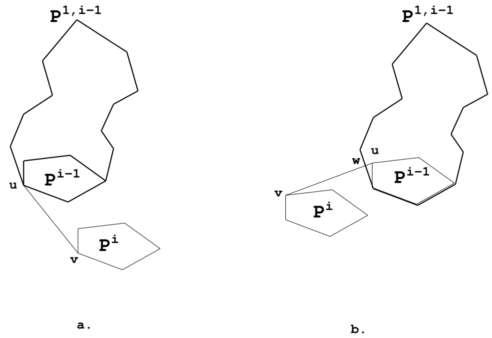

Figure 2. The two cases for adding polygon P^i to $P^{1,i-1}$: a. extreme vertex $u \in LC^{1,i-1}$ and b. extreme vertex $u \notin LC^{1,i-1}$.

8: find edges $\overline{uv} \in P^{i-1,i}$ and $\overline{u'v'} \in P^{1,i-1}$ such that $\overline{uv} \cap \overline{u'v'} \neq \emptyset$
9: $w \leftarrow \overline{uv} \cap \overline{u'v'} \neq \emptyset$
10: remove vertices $v', \ldots, v_b^{1,i-1}$ from $LC^{1,i-1}$
11: $LC^{1,i} \leftarrow LC^{1,i-1} \cup w, \ldots, v_b^i$
12: **end if**
$\{$Update the right chain of $P^{1,i-1}\}$
$\{$This case is similar to updating to updating the left chain and therefore omitted$\}$
endfunction

Next we prove the correctness of the function. First we show that if $u \notin P^{1,i-1}$ then $LC^{1,i-1} \cap LC^{i-1,i}$ at a single vertex.

Lemma 4.1 *Given a convex polygon P and a monotonically decreasing chain C in the y-direction, if $u \notin P^{1,i-1}$, where u is the left extreme edge of P^{i-1} then $LC^{1,i-1}$ intersects the $LC^{i-1,i}$ only once.*

Proof: The proof is given by contradiction. Assume that $LC^{1,i-1}$ and $LC^{i-1,i}$ intersect more than once. Let edges $\overline{qr}$ and $\overline{st}$ be the edges that $\overline{vu}$ intersects $LC^{1,i-1}$ for the first time (enters $P^{1,i-1}$) and for the second time (leaves $P^{1,i-1}$) respectively (see Figure 3).

Edge $\overline{st} \in LC^{1,i-1}$ was added when polygon $P^k, k < i-1$ was added to $LC^{1,k-1}$. Thus, $\overline{st}$ is either an edge of P^k or an extreme edge of $CH(P^{k-1}, P^k)$. Let L denote a line through $\overline{st}$. Thus, P^k lies below L.

Edge $\overline{uv}$ intersects $\overline{st}$ and therefore $u \in P^{i-1}$ is above L. This implies that P^{i-1} is above P^k contradicts to the assumption that C is a monotonically decreasing chain since $k < i-1$. $\square$

We can now show that the function correctly adds P^i to $P^{1,i-1}$.

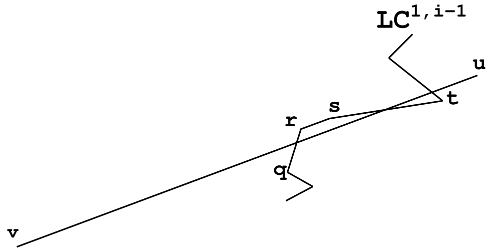

Figure 3. An impossible configuration that occurs if $LC^{1,i-1}$ would intersect $LC^{i-1,i}$ more than once.

Lemma 4.2 *Given a convex polygon $P = \{v_1, \ldots, v_n\}$ and a monotonically decreasing chain $C = \{c_1, \ldots, c_m\}$ in the y-direction, the resulting polygon, which is constructed by adding polygon P^i to $P^{1,i-1}$ by Function AddPolygon, is the path polygon $P^{1,i}$.*

Proof: The proof is given by showing: a. the output polygon is monotone in the y-direction; and b. the output polygon is the minimum size polygon that covers the regions occupied by P as it is translated along C.

Case 1 Output polygon is monotone: for brevity we show that the constructed left chain is monotone in the y-direction. The algorithm accepts as input $P^{1,i-1}$ and therefore $LC^{1,i-1}$ is monotone. The algorithm finds as a first step polygon $Q = CH(P^{i-1}, P^i)$. From Observation 3 there are two options for uniting $LC^{i-1,i}$ and $LC^{1,i-1}$. First option occurs when $u \in LC^{1,i-1}$ where u is that the left extreme edge of P^{i-1}. Here, $LC^{1,i}$ is monotone because the sub-chain of $LC^{1,i-1}$ consisting of $\{v_t^{1,i-1}, \ldots, u\}$ is monotone and the concatenated sub-chain of $LC^{i-1,i}$ consisting of $\{u, \ldots, v_b^{i-1,i}\}$ is monotone. This case is handled in lines 5,6 of function AddPolygon.

The second option is that $u \notin LC^{1,i-1}$ where u is the left extreme edge of P^{i-1}. Here, by Lemma 4.1 there is an intersection point w such that the sub-chain of $LC^{1,i-1}$ consisting of $\{v_t^{1,i-1}, \ldots, w\}$ is monotone and the concatenated sub-chain of $LC^{i-1,i}$ consisting of $\{w, \ldots, v_b^{i-1,i}\}$ is monotone. This case is handled in lines 8-11 of function AddPolygon.

Case 2 Output polygon is the minimum size polygon containing P: the input polygon $P^{1,i-1}$ is a path polygon and therefore the minimum size polygon that covers the regions occupied by P as it is translated along sub-chain $c_1, \ldots, c_{i-1}$. From Lemma 1 $P^{i-1,i}$ is a path polygon and therefore the minimum size polygon that covers the regions occupied by P as it is translated along $\overline{c_{i-1}c_i}$. Therefore the union of the two path polygons is the minimum size polygon that covers the regions occupied by P as it is translated along sub-chain $c_1, \ldots, c_{i-1}$. Function AddPolygon finds the union of the two polygons by determining the left and right chains of the output polygon. □

During its execution, the function AddPolygon adds new vertices to the output path polygon $P^{1,i}$. However, the function also removes vertices from the polygon. Although, the total number of vertices that can be added is $O(n)$ by Corollary 3.1 the number of vertices in $P^{i-1,i}$ grows by constant number. Recall that the function removes the same number of vertices. Thus, we obtain the following corollary.

Corollary 4.1 *The total number of vertices in path polygon $P^{1,i}$ can be nm.*

The time complexity of the function is given next.

Lemma 4.3 *Function AddPolygon requires $O(\log n + \log n \log(nm) + n)$ time in the worst case to add polygon P^i to $P^{1,i-1}$, where n is the number of vertices in P^i.*

Proof:

Finding the $CH(P^{i-1}, P^i)$ can be done in $O(\log n)$ time (note that P^{i-1} and P^i are two copies of the same polygon and the supporting lines are parallel to $\overline{c_{i-1}c_i}$. Finding the intersection between $LC^{1,i-1}$ and $LC^{i-1,i}$ can be done in $O(\log n \log l)$ time similarly to merging two convex hulls, where l is the number of vertices in $P^{1,i-1}$. By Corollary 4.1 the number of vertices in $P^{1,i-1}$ is bounded by nm which yields a time complexity of $O(\log n \log(nm)$. Last the function adds new vertices to $LC^{1,i-1}$ and $RC^{1,i-1}$. In the worst case all the vertices of P^i must be added to $LC^{1,i-1}$ and $RC^{1,i-1}$, which is linear. □

In this section we showed how a single polygon is added to an existing path polygon. Next we present the algorithm that computes the complete path polygon.

4.2 Main Algorithm

In this section we present the main algorithm which computes the complete path polygon $P^{1,m}$

Function CmputePathPolygon(C,P,Q)
  {$C = \{c_1, \ldots, c_m\}$ is a monotone chain in the y-direction}
  {P is the input convex polygon}
  {Q is the output path polygon}
  $P^{1,1} \leftarrow P^1$;
  for $i \leftarrow 2$ to m **do**
    compute polygon P^i
    Call function AddPolyon($P^{1,i-1}$,P^i)
  end for
  $Q \leftarrow P^{1,m}$

Next we show that the algorithm finds the path polygon $P^{1,m}$ and discuss its time complexity.

Lemma 4.4 *Given a convex polygon $P = \{v_1, \ldots, v_n\}$ and a monotonically decreasing chain $C = \{c_1, \ldots, c_m\}$ in the y-direction, the function ComputePathPolygon correctly computes the path polygon $P^{1,m}$.*

Proof:

We show it by induction on the size of the monotone chain C.

Base case: C consists of one vertex. In this case the function ComputePathPolygon returns P^1.

The induction hypothesis is: assume that function ComputePathPolygon correctly computes the path polygon $P^{1,k}$ for a monotone chain C with k vertices.

We show that it correctly computes the path polygon for a monotone chain C with $k+1$ vertices.

At the $k+1$ iteration the function invokes AddPolygon with the path polygon $P^{1,k}$ and P^{k+1}. From the induction hypothesis polygon $P^{1,k}$ was correctly computed by ComputePathPolygon.

By Lemma 4.2 the function AddPolygon correctly adds polygon P^{k+1} to $P^{1,k}$ to form $P^{1,k+1}$. □

The time complexity of the ComputePathPolygon is given next.

Theorem 4.1 *Given a convex polygon $P = \{v_1, \ldots, v_n\}$ and a monotonically decreasing chain $C = \{c_1, \ldots, c_m\}$ in the y-direction, the function ComputePathPolygon computes the path polygon $P^{1,m}$ in $O(mn + m \log n \log(nm))$ time.*

Proof:

The function ComputePathPolygon $m-1$ polygons to an initial path polygon $P^{1,1}$. At each iteration the algorithm it computes the next P^i at a cost $O(mn)$. The function also invokes the function AddPolygon $m-1$ times at a cost of $O(\log n + \log n \log mn + n)$ time.

□

When the size of the monotone chain C is O(n), we obtain the following corollary:

Corollary 4.2 *Given a convex polygon $P = \{v_1, \ldots, v_n\}$ and a monotonically decreasing chain $C = \{c_1, \ldots, c_m\}$ in the y-direction, the function ComputePathPolygon computes the path polygon $P^{1,m}$ in $O(mn + m \log^2 n)$ time.*

5 Conclusions

Reducing the damage to healthy tissue is very important and at times crucial to successful recovery of cancer patients who are treated by radiation therapy. Oncologists often prescribe a large area to be treated in order to ensure that the whole tumour is receiving the prescribed dose. This is especially critical in cases where the tumour moves during treatment e.g., as a result of breathing. In this paper we presented a solution for computing the area covered by a tumour as its moves during treatment. Our solution takes $O(mn + m \log n \log(nm))$ time. We are currently working on pre-clinical tests to incorporate our techniques in the treatment cycle. We are also working on a 3D variation of this problem.

6 Acknowledgements

The authors wish to thank the Natural Sciences and Engineering Research Council of Canada for their support in conducting this research.

References

[1] Stanley Benedict. Looking into patient positioning and organ motion. Slides from the website: www.acmp.org/meetings/hershey-2001/highlights/bendict.pdf, 2001.

[2] Jilan Chen, Mark Lanthier, Miller MacPherson, and Doron Nussbaum. Realtime adjustments to radiation treatment plans due to tumour motion. Manuscript in preparation, 2004.

[3] Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. *Computational geometry: algorithms and applications*. Springer-Verlag New York, Inc., 1997.

[4] A. Ezz, P. Munro, A.T. Porter, J. Battista, D.A. Jaffray, A. Fenser, and S. Osborne. Daily monitoring and correction of radiation field placement using a video-based portal imaging system: a pilot study. *Int. J. Rad. Oncol. Biol. Phys.*, 22:159, 165, 1992.

[5] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, Reading, MA, 2 edition, 1990.

[6] E.C. Ford, G.S. Mageras, E. Yorke, K.E. Rosenzweig, R. Wagman, and C.C. Ling. Evaluation of respiratory movement during gated radiotherapy using film and electronic protal imaging. *Int.J.Radiation Oncology Biol.Phys.*, 52(2):522,531, 2002.

[7] Michel G, Herman, Kevin O. Khadivi, Lawrence Kleinberg, Irene Gage, and Ross A. Abrams. Effects of respiration on target and critical structure positions during treatment assessed with movie-loop electronic portal imaging. *Int.J.Radiation Oncology Biol.Phys.*, 39:163, 1997.

[8] L.H. Gerig, S.F. E1-Hakin, J. Szanto, D. Salhani, and A. Girard. The development and clinical application of a patient position monitoring system. In *In Videometrics iii*, SPIE 2350, Boston, MA, Nov 1994. SPIE.

[9] Scott W. Hadley. *Video-based patient positioning for external beam radiation therapy*. Doctor of philosophy, The University of Chicago, Chicago, Illinois, 2002.

[10] J.H. Killoran, H.M. Kooy, D.J. Gladstone, F.J. Welte, and C.J. Beard. A numerical simulation of organ motion and daily setup uncertainties: Implications for radiation therapy. *Int.J.Radiation Oncology Biol.Phys.*, 37(1):213,221, 1997.

[11] John R. Van Sornsen De Koste, Hans C.J. De Boer, Regine H. Schuchhard-Schipper, Suresh Senan, and Ben J. M. Heijmen. Procedures for high precision setup verification and correction of lung cancer patients using ct-simulation and digitally reconstructed radiographs (drr). *Int.J.Radiation Oncology Biol.Phys.*, 55(3):840,810, 2003.

[12] Hideo D Kubo and Bruce C Hill. Respiration gated radiotherapy treatment: a technical study. *Phys. Med. Biol.*, 41:83, 91, 1996.

[13] A.L. Mckenzie. How should breathing motion be combined with other errors when drawing margins around clinical target volumes? *The British Journal of Radiology*, 73:973,977, 2000.

[14] Varian Medical Systetems. Delivery systems: Respiratory gating : Real-time position management (rpm). Internet Communication: www.varian.com/onc/prd057.html, 2003.

[15] J.W. Wong, M.B. Sharpe, D.A. Jaffray, J.M. Robertson, J.S. Stromberg, V.R. Kini, and A.A. Nartinez. The use of active breathing control (abc) to reduce margin for breathing motion. *Int.J.Radiation Oncology Biol.Phys*, 44(4):911,919, 1999.