Phys. Med. Biol. **54** (2009) 569–589 doi:10.1088/0031-9155/54/3/007

A novel unidirectional intensity map segmentation method for step-and-shoot IMRT delivery with segment shape control

J M Artacho, X Mellado, G Tobías, S Cruz and M Hernández

Communications Technology Group, Aragon Institute on Engineering Research,
University of Zaragoza, Spain

E-mail: jartacho@unizar.es, xme@unizar.es, cruzll@unizar.es and mhg@unizar.es

Received 6 June 2008, in final form 16 November 2008
Published 6 January 2009
Online at stacks.iop.org/PMB/54/569

Abstract

In intensity modulated radiation therapy (IMRT), intensity maps are computed from prescribed doses to target volumes, adding dose restrictions to the surrounding tissues. Those intensity (fluence) maps are discretized into matrices of natural numbers and translated to sequences of multileaf collimator (MLC) leaf movements, which will finally deliver the computed x-ray intensities. A unidirectional leaf sequencing algorithm that controls the shape of the segments and reduces leaf motion time for step-and-shoot dose delivery is presented. The problem of constructing segments in controlling its shape was solved by synchronizing right leaves motion. This is done without increasing the number of segments, or the total number of monitor units, and taking into account the unidirectional leaf motion and the interleaf collision constraints. The method was tested using random matrices and a clinical case planned with the PCRT 3D® planning system. Compared to other unidirectional leaf sequencing methods, the proposed algorithm performs very similarly. But, in addition, the segment shape control produces segments with smoother outlines and more compact shapes, which may help to reduce MLC-specific effects when delivering the planned fluence map. Finally, as a result of imposing unidirectionality, this algorithm can be extended for sliding window segment generation.

1. Introduction

Intensity modulated radiation therapy (IMRT) is an external radiotherapy technique for cancer treatment. A computer-controlled linear accelerator equipped with an MLC is used to deliver precise radiation doses of x-rays to a malignant tumour, or specific areas within a tumour. The radiation dose is designed to conform to the three-dimensional shape of the tumour,

0031-9155/09/030569+21$30.00

as a summation of individual non-uniform intensity radiation patterns from different angles (Artacho *et al* 2007, Lorente *et al* 2006).

An MLC contains two opposite banks of metal leaves. For each row, there is one leaf located to the left and another located to the right. Those leaves may move inwards or outwards to cover spatial positions, shaping the beam. It can be used in static mode (SMLC) (Galvin *et al* 1993) or step-and-shoot, and in dynamic mode (DMLC) or sliding window (van Santvoort and Heijmen 1996).

The SMLC mode consists of discrete steps, the leaves do not move while the beam is on, whereas in the DMLC mode, the beam is on while the leaves are continuously moving at variable speed. In the SMLC mode, the non-uniform intensity radiation patterns are discretized into fluence matrices, whose elements are naturals. The beam produced by a linear accelerator is uniform. Thus, a segmentation method is needed for delivering these non-uniform patterns, since it sequences the fluence matrix in different shaped beams (segments) with different beam weights (Kalinowski 2006).

The MLC segmentation problem is a difficult combinatorial problem. The optimal solution is the one which minimizes the whole treatment time, and the solution is not unique. There are three factors in this treatment time: the total beam-on time or monitor units (MU), the verification and recording cycle (V&R), and the MLC leaf motion time, which is directly related to the number of segments (NS). The delivery time is approximately equal to the total number of monitor units (TNMU), plus the bigger of the other two factors (Siochi 1999).

There are many SMLC segmentation algorithms (Galvin *et al* 1993, Bortfeld *et al* 1994, Xia and Verhey 1998, Siochi 1999, Dai and Zhu 2001, Crooks *et al* 2002, Kalinowski 2006). Each one has different properties and performance but, in general, a segmentation method is designed to minimize only one factor MU or NS, while keeping the other one at a reasonable value. The V&R factor is usually not taken into account because it depends on the MLC model used. It should be noted that achieving the optimal solution for the target factor is not guaranteed.

1.1. MLC-specific effects

In general, the algorithms published assume that an MLC can deliver exactly the desired intensity map as is, without considering the MLC-specific effects such as the head scattering or the leaf transmission. This assumption can lead to significant discrepancies between the desired and the delivered intensity maps (Hansen *et al* 1998, LoSasso *et al* 1998, Cho and Marks 2000, Seco *et al* 2001, Azcona *et al* 2002).

There are several published solutions for DMLC segmentations of this problem (Convery and Webb 1998, Dirkx *et al* 1998). In these works, leaf transmission, collimator scatter and tongue-and-groove effects are considered, because the input intensity map is modified according to the difference between the desired and the delivered maps, including previous MLC-specific effects. These methods were designed for DMLC, and they cannot be extended to SMLC, as pointed out by Yang and Xing (2003), which proposes an equivalent solution adapted to the static mode. These solutions have a serious drawback, because the modelling and verification of MLC-specific effects is quite difficult and expensive.

On the other hand, two new methods or ways of planning doses for step-and-shoot IMRT were published: the direct aperture optimization (DAO) (Shepard *et al* 2002) and the direct machine parameter optimization (DMPO) (Hardemark *et al* 2003). Both articles discuss the problem of considering the optimization and the delivery (segmentation) as separated problems. This approach causes the differences between the desired and the delivered maps, because the MLC-specific effects cannot be included in the optimization. The proposed solution in both cases is to merge the optimization and the segmentation steps into a single one. This solution is probably the best one, but it has the drawback of coupling the optimization and the segmentation. Thus, it is not valid for conventional IMRT planning systems without completely changing their implementation.

1.2. The segment shape constraint

The approach adopted in this work is based on the results and conclusions of Hansen *et al* (1998), LoSasso *et al* (1998), Budgell *et al* (2000) and Cho and Marks (2000). The use of a large number of segments with complex shapes can increase collimator artefacts. In this situation, there are usually segments with small fields (or unbalanced X–Y axis) and low number of MUs. This is a problem for accurate dose calculation. The output for these segments must be carefully computed and corrected by the dose calculation algorithm, considering the MLC-specific effects. Therefore, those segments introduce tough requirements for geometric accuracy of the MLC and dosimetric accuracy of the linear accelerator.

The number of segments or their MUs are not subject to changes, unless the DAO or the DMPO approaches are used, because the traditional leaf sequencing algorithms cannot fix the NS or constrain the MUs (the MU value directly depends on its segment, so the segment computation should consider the eventual associated weight as a new restriction). However, the segment shape can be influenced imposing a constraint for leaf synchronization. This leaf synchronization can be controlled by the algorithm, in order to balance shape uniformity versus NS or TNMU increase regarding the original solution.

1.3. Summary

The contribution of this work is a SMLC segmentation method incorporating: (1) unidirectionality (Bortfeld *et al* 1994, Siochi 1999) for reducing the leaf motion time, (2) the interleaf collision constraint, so it can be used in MLCs with motion constraints, (3) a leaf synchronization constraint for controlling segment shape, generating segments with smoother outlines and more compact shapes and (4) two different criteria for minimizing either the NS or the TNMU, opposite to the single criterion usually available on other algorithms. The first criterion is a new one proposed in this work for the reduction of NS, and the second one is described in Engel (2005) and Kalinowski (2006) and it obtains segmentations with the optimal TNMU.

2. Method

Usually, a segmentation method decomposes a fluence matrix in different segments plus weights on an iterative process, and each iteration can be divided into two steps. The first step is the computation of a segment (matrix of ones and zeros, understood as a mask) for a given fluence matrix, using a set of constraints. The second step is the computation of the weight associated with the obtained segment, following only one fixed criterion to minimize the NS or the TNMU. Finally, the segment multiplied by the weight is subtracted from the fluence matrix, generating a residual matrix that will be the input fluence matrix for the next iteration.

The proposed algorithm allows the user, at the beginning of the process, to select which criteria will be used, in order to minimize the NS or the TNMU depending on the desired target. The pseudocode in appendix A2 illustrates this process.

This section will be divided into three subsections explaining preliminary definitions, the computation of segments (S_k) and the computation of weights (α_k).

2.1. Definitions and notation

A similar notation and definitions as given by Kalinowski (2006, 2004) and Engel (2005) is used.

Definition 1. *Let A be a natural number matrix with M rows and N columns representing a given fluence matrix. Let S be a segment; a segment is a matrix with the same dimensions as its fluence matrix A but formed only by* [0, 1] *naturals. When a given position in the segment is equal to* 0*, it means this position is covered by a leaf. When it is equal to* 1*, it means this position is letting pass radiation. The segmentation (or decomposition) of A is expressed as*

$$A = \sum_{k=1}^{NS} \alpha_k \cdot S_k, \tag{1}$$

where $\alpha_k > 0$ *is the weight accounting for a relative beam on time for the kth segment. Those weights are directly proportional to the MU to be delivered. NS, as mentioned before, is the number of segments.*

Definition 2. *Let* A_k *be the residual matrix obtained when* $\alpha_{k-1} \cdot S_{k-1}$ *is subtracted from* A_{k-1}*, i.e.,*

$$\begin{aligned} A_1 &= A \\ A_2 &= A_1 - \alpha_1 \cdot S_1 \\ &\vdots \\ A_k &= A_{k-1} - \alpha_{k-1} \cdot S_{k-1} \\ \overrightarrow{0} &= A_k - \alpha_k \cdot S_k. \end{aligned}$$

Definition 3. *In order to simplify the notation, the A matrix is expanded assuming that two zero rows and two zero columns are added at the boundaries,*

$$\begin{array}{lll} \text{Rows:} & a_{0,j} = a_{M+1,j} = 0 & j \in [1, \ldots, N] \\ \text{Columns:} & a_{i,0} = a_{i,N+1} = 0 & i \in [1, \ldots, M]. \end{array}$$

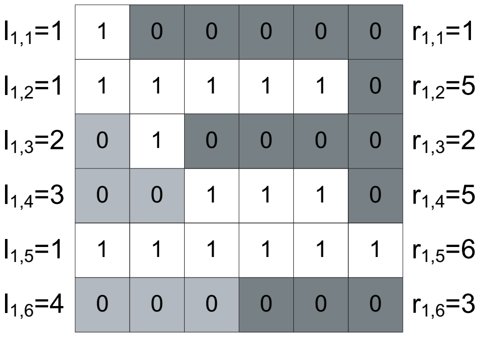
Definition 4. *Let* $l_{k,i}$ *and* $r_{k,i}$ *denote the position of the left and right leaves at the ith row in the kth segment,*

$$1 \leqslant l_{k,i} \leqslant r_{k,i} + 1 \leqslant N + 1 \qquad k \in [1, \ldots, NS], \qquad i \in [1, \ldots, M],$$

where positions $l_{k,i}$ *to* $r_{k,i}$ *are 'opened' and exposed to radiation, while the left leaf at positions* $[0, \ldots, l_{k,i} - 1]$ *and right leaf at positions* $[r_{k,i} + 1, \ldots, N + 1]$ *are blocking radiation. The case of a row filled with zeros (totally closed) is included because it is allowed that* $l_{k,i} = r_{k,i} + 1$*. Figure 1 shows* $l_{k,i}$ *and* $r_{k,i}$ *values for an example segment.*

Definition 5. *In a given row, there is a local peak at column p if both columns* $p - 1$ *and* $p + 1$ *have a lower value. When a set of contiguous repeated numbers is found, only the first one is taken into account and repetitions are ignored. Note that using definition 4, columns* 0 *and* $N + 1$ *are filled with zeros, and they must be taken into account. Table 1 shows some examples of peak detections. The number of local peaks in the ith row can be expressed as*

$$\begin{aligned} &\text{peaks}_i(A) = |\{x \in \mathbb{N} : \exists p \in [1, \ldots, N], \exists q \in [2, \ldots, N+1] : \\ &x = q \wedge p < q \wedge \\ &A(i, p-1) < A(i, p) \wedge A(i, q) < A(i, p) \wedge \\ &\forall\, s : p < s < q : A(i, p) = A(i, s)\}\}.| \end{aligned} \tag{2}$$



$l_{1,1}$=1	1	0	0	0	0	0	$r_{1,1}$=1
$l_{1,2}$=1	1	1	1	1	1	0	$r_{1,2}$=5
$l_{1,3}$=2	0	1	0	0	0	0	$r_{1,3}$=2
$l_{1,4}$=3	0	0	1	1	1	0	$r_{1,4}$=5
$l_{1,5}$=1	1	1	1	1	1	1	$r_{1,5}$=6
$l_{1,6}$=4	0	0	0	0	0	0	$r_{1,6}$=3

Figure 1. Values for $l_{k,i}$ and $r_{k,i}$ in a hypothetical segment. In light grey left leaves and in dark grey right leaves are shown.

Table 1. Local row peak examples. The underlined elements are the peaks.

1 peak	2 peaks
$(1\ \underline{3}\ 2\ 1)$	$(1\ \underline{2}\ 1\ \underline{2}\ 2)$
$(1\ \underline{2}\ 2\ 1)$	$(1\ \underline{3}\ 2\ 3\ \underline{4})$
$(1\ 2\ 2\ \underline{3})$	$(\underline{3}\ 2\ \underline{4}\ 2\ 1)$
$(\underline{5}\ 4\ 3\ 1)$	$(\underline{3}\ 3\ 2\ \underline{3}\ 3)$

2.2. *Segment computation* (S_k)

The generation of the S_k segment is driven by a set of constraints applied at the same time on the A_k fluence matrix. Once the segment has been computed, a routine for solving collisions not predictable by the ICC constraint is used, and the eventual segment is obtained.

2.2.1. Basic constraints. The leaf-sequencing algorithm is designed to meet two basic constraints: unidirectionality and interleaf collision constraint. For simplicity it is assumed that the leaf motion is from left to right, but it is straightforward to reverse the direction.

Unidirectionality. This ensures that no new maxima would be created in the ith row, and therefore, the right leaf would not move backwards (from right to left),

$$S_k(i,j) = \begin{cases} 1 & \text{if } \forall\, x \in [1,\ldots,j-1] : A_k(i,j-x) \leqslant A_k(i,j) \\ 0 & \text{otherwise} \end{cases}$$
$$i \in [1,\ldots,M],\ j \in [1,\ldots,N]. \tag{3}$$

Example 1. Consider the linear decomposition of a little test matrix using only this constraint.

$$\underbrace{\begin{pmatrix} 3\ 1\ 1 \\ 1\ 1\ 2 \end{pmatrix}}_{A} = \underbrace{\begin{pmatrix} 1\ 0\ 0 \\ 1\ 1\ 1 \end{pmatrix}}_{S_1} + \underbrace{\begin{pmatrix} 1\ 0\ 0 \\ 0\ 0\ 1 \end{pmatrix}}_{S_2} + \underbrace{\begin{pmatrix} 1\ 1\ 1 \\ 0\ 0\ 0 \end{pmatrix}}_{S_3}.$$

This constraint can be understood as a single row sliding window (or queue) that moves from left to right for each row independently. It adds (pushes) a new position to the window front only if it has equal or greater intensity than those currently inside. It removes (pops) the last position at the window back if it has intensity 0. More than one position can be added or removed at the same time. Note that unidirectionality does not prevent collisions; the second mask of example 1 is a clear example.

Interleaf collision constraint. This does not allow the overlapping of opposite leaves in adjacent rows, which is basically a dependence among adjacent sliding windows,

$$S_k(i,j) = \begin{cases} 1 & \text{if } \forall\, x \in [1,\ldots,j-1] : A_k(i \pm 1, j-x) - x + 1 \leqslant A_k(i,j) \\ 0 & \text{otherwise} \end{cases}$$
$$i \in [1,\ldots,M], j \in [1,\ldots,N]. \tag{4}$$

Example 2. In order to illustrate the meaning of this equation, it is applied on the little example together with unidirectionality. The upper row is the fluence matrix decomposed in segments and weights (no weight means it is equal to 1). The lower row is the residual matrix associated with each segment,

$$\begin{pmatrix}3&1&1\\1&1&2\end{pmatrix} = \begin{pmatrix}1&0&0\\1&0&0\end{pmatrix} + \begin{pmatrix}1&0&0\\0&0&0\end{pmatrix} + \begin{pmatrix}1&1&1\\0&1&1\end{pmatrix} + \begin{pmatrix}0&0&0\\0&0&1\end{pmatrix}$$
$$\begin{pmatrix}2&1&1\\0&1&2\end{pmatrix} \to \begin{pmatrix}1&1&1\\0&1&2\end{pmatrix} \to \begin{pmatrix}0&0&0\\0&0&1\end{pmatrix} \to \begin{pmatrix}0&0&0\\0&0&0\end{pmatrix}.$$

Equation (4) always ensures ICC. Although, in some cases, this restriction can be relaxed preserving ICC while reducing the NS. The relaxation is only for the mathematical constraint just as it is formulated, because sometimes it is too restrictive, but the real constraint will always be fulfilled. The softening is done adding a new variable called ‘subtraction’ and represented by r,

$$S_k(i,j) = \begin{cases} 1 & \text{if } \forall\, x \in [1,\ldots,j-1] \forall r \in [0,\ldots,j] : \\ & \quad A_k(i \pm 1, j-x) - x + 1 - r \leqslant A_k(i,j) \\ 0 & \text{otherwise} \end{cases}$$
$$i \in [1,\ldots,M], j \in [1,\ldots,N]. \tag{5}$$

The initial value of r is set to N. The algorithm decreases its value iteratively as long as the segmentation process finishes without results, because an ICC violation occurs at some point of the segmentation and it is not solvable by the basic collision routine (as explained in section 2.2.3). This minimization continues until the first value (and the highest) of r fulfilling the ICC is found, and the segmentation process ends successfully. If the eventual value of r is equal to 0, this constraint is equivalent to the one described in equation (4), and it means the reduction of the NS is not possible.

The higher the r value is, the lower the NS will be, because r weakens the ICC, allowing the right leaf to advance, even if current fluence value is lower than the compared neighbour ones.

Example 3. To illustrate the effect of equation (5) in the segmentation process, the modified constraint is applied on the example 2 matrix with $r = 1$. The lower row is the residual matrix associated with each segment,

$$\begin{pmatrix}3&1&1\\1&1&2\end{pmatrix} = \begin{pmatrix}1&0&0\\1&0&0\end{pmatrix} + \begin{pmatrix}1&0&0\\0&1&1\end{pmatrix} + \begin{pmatrix}1&1&1\\0&0&1\end{pmatrix}$$
$$\begin{pmatrix}2&1&1\\0&1&2\end{pmatrix} \to \begin{pmatrix}1&1&1\\0&0&1\end{pmatrix} \to \begin{pmatrix}0&0&0\\0&0&0\end{pmatrix}.$$

In this case, when A_2 is segmented, the second row can be included and it yields to a segmentation with one less segment.

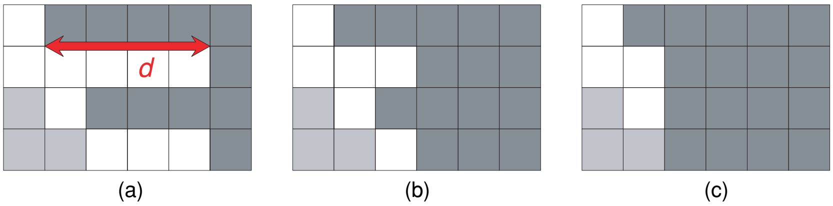


Figure 2. *d* variable on a hypothetical segment: (a) original segment with *d* represented graphically as an arrow, (b) $d \leqslant 2$ and (c) $d \leqslant 1$.

2.2.2. The segment shape constraint. A new variable called 'depth' and represented by d is added. It provides a certain degree of control over the segment shape, by limiting the difference between the adjacent leaves at the right side. As the ICC (section 2.2.1), it is a dependence among adjacent rows. Figure 2 illustrates the effect of d. Using the definition of $r_{k,i}$ previously given in definition 4:

$$|r_{k,i} - r_{k,i\pm 1}| \leqslant d \qquad d \in [0, \ldots, N]. \tag{6}$$

The initial value of d is set to 0 and the algorithm maximizes it iteratively under the condition of keeping a maximum NS or TNMU according to the selected criterion. To be precise, when the subtraction value has been optimized, the segmentation process starts again with the subtraction value fixed and varying d. If the segmentation process fails with $d = n, n \in [0, \ldots, N]$, it is repeated with $d = n + 1$, because the lower the d value is, the smoother is the segment outline, but the higher is its NS or TNMU. This iterative process continues until the first value of d that meets the maximum NS or TNMU is found. Section 2.3.4 explains in detail the relation between the d variable and the NS and TNMU. The pseudocode in appendix A1 formalizes this process.

2.2.3. General interleaf collision detection and solving. The ICC introduced in section 2.2.1 can be understood as a way of synchronizing the advance of the queues generated by the unidirectional constraint. The proposed constraint ensures that every single row is synchronized with its adjacent ones, but this is not enough to ensure no ICC violation on the whole segment. Thus, a routine for a global check is needed. Let A be an example fluence matrix, and $\alpha_1 \cdot S_1$ its segmentation,

$$A = \begin{pmatrix} 0\,0\,0\,5\,5 \\ 0\,0\,0\,0\,0 \\ 5\,5\,0\,0\,0 \end{pmatrix} = 5 \cdot \begin{pmatrix} 0\,0\,0\,1\,1 \\ 0\,0\,0\,0\,0 \\ 1\,1\,0\,0\,0 \end{pmatrix} = \alpha_1 \cdot S_1.$$

Segment S_1 fulfils the proposed unidirectionality and interleaf collision constraints, but the algorithm faces an unsolvable conflict if only these rules are applied. Therefore, these situations should be fixed using an algorithm like the following one: (1) find the row whose right leaf is behind any other right leaf in the segment, i.e. the smallest $r_{k,i}$, in the case of draw, the row whose left leaf has the smallest $l_{k,i}$ is chosen, (2) use previous row index as the starting point for iterating with decreasing row indices (upwards) applying equation (7). Then, the same procedure is applied with increasing row indices (downwards) applying equation (8). At the end, any row violating the ICC must be closed,

$$\text{Upwards: if } r_{k,i} < (l_{k,i-1} - 1) \rightarrow l_{k,i-1} = r_{k,i} + 1;\ r_{k,i-1} = r_{k,i}; \tag{7}$$

$$\text{Downwards: if } r_{k,i} < (l_{k,i+1} - 1) \rightarrow l_{k,i+1} = r_{k,i} + 1;\ r_{k,i+1} = r_{k,i}. \tag{8}$$

2.2.4. The tongue-and-groove constraint. The tongue-and-groove design of the MLCs causes an underdose effect in a narrow band at the overlapping region between two adjacent rows. This effect can be removed from the segmentation methods introducing the tongue-and-groove constraint (TGC) (van Santvoort and Heijmen 1996, Siochi 1999, Kalinowski 2006).

The TGC also smoothes segment outlines and compact segment shapes, so it can be used in a similar way as the segment shape constraint proposed in section 2.2.2. However, this constraint increases, in average, the NS and the TNMU (van Santvoort and Heijmen 1996, Kalinowski 2004) and the increase cannot be controlled, as it is done with the proposed shape constraint in section 2.3.4.

For the ONS and the OTNMU algorithms, the TGC can be introduced following the same formulation described in Kalinowski (2004).

$$\begin{aligned}&A(i,j)\leqslant A(i+1,j)\wedge S(i,j)=1\Rightarrow S(i+1,j)=1:i\in[1,\ldots,M-1],\ j\in[1,\ldots,N]\\&A(i,j)\leqslant A(i-1,j)\wedge S(i,j)=1\Rightarrow S(i-1,j)=1:i\in[2,\ldots,M],\ j\in[1,\ldots,N].\end{aligned}\tag{9}$$

Equation (9) ensures that if a fluence value is smaller than its column neighbours ($i+1$ or $i-1$), it should be exposed at the same time as them. Thus, it can be guaranteed that the tongue-and-groove region, at least, receives the smaller dose.

2.3. Weight computation (α_k)

The α_k weight associated with the S_k segment is generated using a criterion for minimizing the NS or the TNMU. The proposed method allows us to select which criterion will be used before the segmentation process starts, and it cannot be changed during the execution. A novel criterion for the minimization of the NS is proposed and the criterion for obtaining optimum TNMU is taken from Kalinowski (2006) and Engel (2005).

First, the general strategy for computing weights is explained, and then each criterion is explained in detail.

2.3.1. Minimization criteria. The selection of one criterion or another would depend on the desired factor to be minimized, the NS or the TNMU. They are used in the computation of the α_k weight associated with the S_k segment. On both criteria, the procedure to obtain the candidate weights and search for the best one is the same:

(i) The kth candidate weights are obtained as the naturals in the range between 1 and the minimum fluence value (represented by ω) in the kth fluence matrix when the kth mask is superimposed. Alternatively, ω can be understood as the lowest fluence value among the $l_{k,i}$ in opened leafs,

$$\begin{aligned}\omega=x\in\mathbb{N}:&\{\exists i\in[1,\ldots,M]:l_{k,i}\leqslant r_{k,i}\wedge x=A_k(i,l_{k,i})\}\\&\wedge\{\forall p\in[1,\ldots,M]:(l_{k,p}\leqslant r_{k,p})\Rightarrow x\geqslant A_k(p,l_{k,p})\}\\&k\in[1,\ldots,NS].\end{aligned}\tag{10}$$

Let Ω be the set with all the fluence values between 1 and ω,

$$\Omega=\{x\in\mathbb{N}:1\leqslant x\leqslant\omega\}.\tag{11}$$

(ii) The mask is multiplied with all the weights in the Ω set and subtracted from the current fluence matrix, so as to compute a set pair: weight and its residual matrix,

$$\forall\,\beta\in\Omega:A_{k+1}^{\beta}=A_k-\beta\cdot S_k.\tag{12}$$

(iii) Lastly, for each pair, the criterion selected is applied and if the current pair is better than previous ones, its candidate weight becomes the new temporary α_k weight.

Intuitively, the best choice is the highest weight in the Ω set, because it is the one delivering the larger dose. However, this way of proceeding is not the best, because smaller values may reduce better the value heterogeneity of the fluence matrix, facilitating subsequent segmentations and consequently achieving faster segmentations than a ‘greedy’ approach.

2.3.2. Optimizing NS (ONS). The objective is to minimize the number of segments (NS) and note that the optimal NS does not imply the optimal TNMU.

The weight α at iteration k is the natural value reducing the biggest number of row local peaks at A_k, i.e. A_{k+1} will tend to have fewer peaks than A_k. The idea behind this criterion is that the fewer peaks in a row, the faster it is segmented. Thus, the fewer row local peaks in a matrix, the faster its segmentation is done.

As explained in section 2.3.1 a list of pair: candidate weight and its residual matrix will be computed. For each residual matrix, the total number of local row peaks is computed summing the local peaks of each row,

$$\text{peaks}(A) = \sum_{i=1}^{M} \text{peaks}_i(A). \tag{13}$$

The residual matrix with the fewer peaks will determine the candidate weight that becomes the final α_k,

$$\begin{aligned}\alpha_k = \beta \in \Omega : \{A_{k+1} = A_k - \beta \cdot S_k\}\\ \wedge\big\{\forall\, \gamma \in \Omega : A_{k+1}^{\gamma} = A_k - \gamma \cdot S_k \wedge \text{peaks}(A_{k+1}) \leqslant \text{peaks}\big(A_{k+1}^{\gamma}\big)\big\}.\end{aligned} \tag{14}$$

Example 4. Let $\omega = 2$ be the lowest fluence value for the first segment of a given fluence matrix. Thus, $\Omega = \{1, 2\}$ are the candidate values. After computing $A_2^{\beta=1} = A_1 - (1 \cdot S_1)$ and $A_2^{\beta=2} = A_1 - (2 \cdot S_1)$ the algorithm decides $\beta = 2$ is the best choice, and it becomes the final α_1 value, because $\text{peaks}\big(A_2^{\beta=1}\big) = 27$ and $\text{peaks}\big(A_2^{\beta=2}\big) = 20$.

2.3.3. Optimizing TNMU (OTNMU). The objective is to minimize the following summation:

$$TNMU = \sum_{k=1}^{NS} \alpha_k. \tag{15}$$

The weight α at iteration k is computed in an equivalent way as defined in Kalinowski (2006) and Engel (2005):

(i) The TNMU complexity of a row, denoted as $c_i(A)$, is the optimal TNMU for the ith (single) row segmentation (Kalinowski 2006, Engel 2005),

$$c_i(A) = \sum_{j=1}^{N} \max(0, a_{i,j} - a_{i,j-1}) \qquad i \in [1, \ldots, M], \quad j \in [1, \ldots, N]. \tag{16}$$

(ii) The TNMU complexity of a matrix, denoted as $c(A)$, is the optimal TNMU for its segmentation (Kalinowski 2006, Engel 2005),

$$c(A) = \max_i(c_i(A)) \qquad i \in [1, \ldots, M]. \tag{17}$$

(iii) Using the TNMU complexity property, the algorithm knows in advance the optimal TNMU for a residual matrix. Therefore, the algorithm can use this property to select the candidate weight plus residual matrix with minimum value, achieving the optimal segmentation in terms of MU. Equation (18) formulates the process using previous equation (17) and the Ω set defined in equation (11),

$$\alpha_k = \beta \in \Omega : \{A_{k+1} = A_k - \beta \cdot S_k\} \\ \wedge \big\{\forall\, \gamma \in \Omega : A^{\gamma}_{k+1} = A_k - \gamma \cdot S_k \wedge \beta + c\big(A_{k+1}\big)\big) \leqslant \big(\gamma + c(A^{\gamma}_{k+1})\big)\big\}. \quad (18)$$

Example 5. Using example 4 for the ONS. This time, the algorithm could decide $\beta = 1$ is the best choice, because $c\big(A_2^{\beta=1}\big) = 50$, $c\big(A_2^{\beta=2}\big) = 52$ and therefore $1 + 50 < 2 + 52$.

2.3.4. Segment shape constraint relaxation. Summarizing, the segmentation process described in this work consists in two main steps. First, the segmentation of the fluence matrix is done several times, optimizing iteratively the subtraction value, but not using the shape constraint. Second, another loop of segmentations is executed, using the subtraction as a constant for optimizing the depth value, and improving segment shapes.

The optimization of the depth variable is done in the following way. The eventual NS or TNMU (depending on the criterion selected) of the subtraction optimization is stored as the ‘stop’ value. The lower the depth value, the more synchronized the left leaves will be, but this would also yield bigger NS or TNMU. The depth variable starts set to 0 and the segmentation is carried out imposing this maximum depth. If the segmentation process reaches the stop value without finishing, it means that the depth condition is too restrictive, and it should be increased. This iterative process continues until a value of depth is found that obtains a segmentation with equal NS or TNMU than the stop value. This condition can be relaxed by increasing the stop value by a percentage (equation (19)); therefore, the result will be better in terms of shape, but worse in terms of NS or TNMU,

$$\text{stop} = \text{original} + (\text{original} \cdot \text{percentage}). \quad (19)$$

Example 6. Let the original NS be 22 and the percentage be 10%, which implies that the second step can produce segmentations up to 24 segments. In the first case, without softening the stop value, depth can be 4. In the second case, it may be reduced to 3 or 2.

3. Results

From now on, the proposed algorithm with different criteria will be referred as separate algorithms (OTNMU and ONS) due to the differences found in results and behaviour.

3.1. Compared segmentation methods and settings

The results section will show the performance of OTNMU and ONS against the leaf sequencing methods described in Galvin *et al* (1993), Bortfeld *et al* (1994), Xia and Verhey (1998), Siochi (1999), Kalinowski (2006).

The method published in Siochi (1999) is a combination of two algorithms applied in two steps embedded in an iterative optimization process. The first step is called *extraction*. It is based on Galvin *et al* (1993), and its output is a pure non-unidirectional SMLC segmentation. The second step is called *rod pushing* (RP). It is a geometrical reformulation of the sweep technique described in Bortfeld *et al* (1994), and its output is a unidirectional segmentation. Both methods are combined in an iterative optimization process, which is driven by a formula that measures the treatment time in a realistic way, taking into account the TNMU, the leaf motion time and the V&R.

Only the rod pushing (RP) technique was implemented for comparison, discarding the extraction part. There were two reasons for this decision: (1) the ONS and OTNMU can be compared with another unidirectional method; also, the RP with ICC (and without the tongue-and-groove constraint) is optimum regarding the TNMU (Kalinowski 2006)[p 1016], and (2) there are not enough details in the original paper to reproduce accurately the implementation of the whole optimization process.

Finally, three remarks have to be considered: (1) although (Galvin *et al* 1993, Bortfeld *et al* 1994) algorithms were designed without the ICC, in Xia and Verhey (1998) they were modified to include this constraint and compare in a fair way, because the ICC increases the number of segments by approximately 25% on both algorithms. (2) Unidirectional segmentations with RP and the proposed algorithms are done in both directions, from left to right and from right to left, and the best solution is selected. (3) The results are divided into two groups, depending on the constraints applied. The ICC group only uses this constraint. The TGC group uses the ICC plus the TGC.

3.2. Data and experimental setting

The methodology proposed by Xia and Verhey (1998), also used in Que (1999) and Kalinowski (2006), is followed:

(i) 1000 15 × 15 matrices were segmented, each having random natural values from 0 to L. The algorithms described in Xia and Verhey (1998) and RP were implemented, but only the second one was used in the testing. The results for Galvin *et al* (1993), Bortfeld *et al* (1994), Xia and Verhey (1998), Kalinowski (2004, 2006) were taken from Xia and Verhey (1998), Kalinowski (2004, 2006). This experiment will allow us to compare the proposed algorithm with the others from the statistical point of view.

(ii) A clinical case generated from the PCRT 3D® (Técnicas Radiofísicas, S.L. C/Gil de Jasa, 18E, 50006 Zaragoza, Spain, www.trf.es) planning system, with five coplanar and equiangular beams, is used to compare results between the RP (Xia and Verhey 1998), OTNMU and ONS in a real prostate cancer case. The comparison has two objectives: (1) comparing in a real clinical case that can be substantially different from random generated matrices and (2) comparing with well-known methods; one unidirectional (RP) and another non-unidirectional (Xia and Verhey 1998).

3.3. Experiment with random matrices

Tables 2 and 3 gather the results of the test with random matrices showing the NS and the TNMU, respectively.

3.4. Experiment with a IMRT planning for a real prostate cancer

The fluence matrices obtained for a real case of prostate cancer from the PCRT 3D® treatment planning system are now considered. The tumour is irradiated from five different coplanar and equiangular beams (36°, 108°, 180°, 252° and 324°) in a 72 Gy plan. The dose–volume constraints used were very similar to the described in Memorial Sloan-Kettering Cancer Center. Departments of Medical Physics and Radiology (2003). For the rectum and the bladder, the constraint was 70% of the volume receives $<$ 40% of the prescription dose.

Table 2. Average NS. OT = OTNMU and ON = ONS.

	Non unidirectional				Unidirectional						
	ICC			TGC	ICC				TGC		
L	Gal	Xia	Kal	Kal	Bor	RP	OT	ON	RP	OT	ON
3	13.4	13.3	12.6	15.5	17.7	15.2	15.6	15.6	16.4	16.4	16.4
4	20.4	18.6	14.5	18.0	22.8	19.1	19.7	19.6	20.9	20.8	20.7
5	20.4	19.0	16.0	20.5	27.9	22.8	23.6	23.3	25.2	25.0	24.8
6	21.5	20.3	17.2	22.6	32.8	26.5	27.4	27.0	29.4	29.2	28.8
7	27.1	20.0	18.2	24.3	37.9	30.1	31.4	30.6	33.6	33.3	32.6
8	28.2	24.3	19.1	25.7	42.8	33.7	35.0	33.9	37.7	37.4	36.3
9	28.3	24.3	19.9	27.0	47.8	37.1	38.6	37.1	41.7	41.4	39.9
10	28.9	25.7	20.7	28.3	52.6	40.6	42.3	40.3	45.6	45.3	43.3
11	30.9	25.7	21.3	29.5	57.6	43.9	45.8	43.2	49.4	49.1	46.6
12	34.8	27.0	21.9	30.5	62.4	47.2	49.1	46.2	53.2	52.8	49.8
13	35.5	26.9	22.5	31.4	67.3	50.5	52.5	49.0	56.8	56.5	52.9
14	35.6	26.9	23.0	32.2	72.2	53.6	55.7	51.7	60.3	60.0	56.0
15	35.9	26.7	23.5	33.1	77.1	56.7	58.9	54.6	63.7	63.5	59.0
16	41.7	30.0	24.0	33.9	82.0	59.7	62.1	57.2	67.1	66.8	61.8

Table 3. Average TNMU. OT = OTNMU and ON = ONS.

	Non unidirectional				Unidirectional						
	ICC			TGC	ICC				TGC		
L	Gal	Xia	Kal	Kal	Bor	RP	OT	ON	RP	OT	ON
3	19.7	19.5	15.4	16.6	17.7	15.4	15.7	15.7	16.5	16.5	16.5
4	40.5	29.6	19.5	21.2	22.8	19.6	19.9	19.9	21.1	21.0	21.0
5	40.1	30.9	23.6	25.8	27.9	23.6	24.0	24.1	25.7	25.3	25.4
6	44.2	46.8	27.6	30.3	32.8	27.7	28.1	28.3	30.1	29.7	29.8
7	67.1	45.6	31.7	34.9	37.9	31.7	32.2	32.5	34.6	34.1	34.3
8	72.3	63.4	35.7	39.2	42.8	35.8	36.3	36.7	39.1	38.4	38.8
9	72.3	67.1	39.8	43.6	47.8	39.8	40.3	40.8	43.6	42.7	43.1
10	76.5	68.6	43.8	48.2	52.6	43.8	44.4	45.1	48.1	47.1	47.7
11	81.4	68.6	47.7	52.9	57.6	47.8	48.5	49.2	52.5	51.4	52.1
12	106.8	101.1	51.8	57.2	62.4	51.8	52.5	53.3	57.0	55.7	56.4
13	101.1	100.6	55.7	61.7	67.3	55.8	56.5	57.5	61.4	60.0	60.9
14	112.7	100.0	59.8	66.0	72.2	59.8	60.6	61.6	65.8	64.3	65.4
15	116.0	98.0	63.8	70.6	77.1	63.8	64.6	65.8	70.3	68.6	69.7
16	154.5	124.9	67.7	74.8	82.0	67.8	68.7	69.9	74.7	72.9	74.0

Table 4 shows the NS and the TNMU for the full set of fluence matrix segmentations. OTNMU and ONS use the subtraction variable r, and variable d with $percentage = 0\%$. Thus, the optimization of d is done without increasing the NS or the TNMU originally obtained with the r variable.

3.4.1. Segment shape comparison. In this section, the segment shapes between Xia and Verhey (1998), RP, OTNMU and ONS are compared. The third beam (180°) segmentation from a previous real case is used, because all the unidirectional methods segment it from left

Table 4. NS and TNMU obtained for each beam in a real prostate cancer case.

	NS						TNMU					
	36°	108°	180°	252°	324°	Total	36°	108°	180°	252°	324°	Total
ICC												
Xia	36	13	32	15	31	127	267	108	194	80	214	863
RP	25	13	32	13	28	111	92	86	80	64	106	428
OTNMU	26	14	31	13	29	113	92	86	80	64	106	428
ONS	25	14	28	11	25	103	96	88	126	64	142	516
TGC												
RP	32	15	32	12	33	124	138	86	86	69	135	514
OTNMU	30	26	33	13	33	135	117	86	81	69	111	464
ONS	29	16	31	11	34	121	143	97	95	80	154	569

to right, and it will help to see the differences when comparing. The fluence matrix to be considered is

$$\begin{pmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&40&40&40&40&40&40&0&0&0&0&0&0&30&40&40&40&40&40&40&40&40&0\\
0&40&40&40&40&40&40&40&40&40&40&40&40&40&40&40&40&40&40&40&40&40&0\\
0&40&40&40&40&40&40&40&40&37&22&28&11&40&40&40&40&40&40&40&40&13&0\\
0&40&40&40&40&40&40&32&27&22&25&30&17&36&30&40&40&40&40&40&40&0&0\\
0&0&40&40&40&40&40&40&32&33&20&29&29&37&40&40&40&40&40&40&40&0&0\\
0&0&0&0&40&40&40&40&29&24&31&27&38&40&40&40&40&40&40&0&0&0&0\\
0&0&0&0&0&0&40&40&27&32&18&26&40&40&40&40&40&0&0&0&0&0&0\\
0&0&0&0&0&0&40&40&40&40&39&40&40&40&40&40&40&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&15&40&40&40&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{pmatrix}.$$

First, the segmentations obtained by Xia and Verhey (1998) and RP methods are shown in figure 3, and the segmentations obtained by the ONS and the OTNMU without d are shown in figure 4. If only the r variable is used, the algorithms try to obtain segmentations with less NS but not with better shapes. Thus, it is important to remark that some segment shapes in figure 3 and 4 have:

(i) Quite irregular shapes, e.g. first and second row segments in RP and OTNMU.
(ii) Several disconnected subsegments, e.g. from 16th to 25th segment in RP, OTNMU and ONS.
(iii) Rather small areas, especially in the Xia and Verhey (1998) method with most of the segments being tiny apertures. Although the last ones of the dynamic methods are relatively small as well.

Second, figure 5 illustrates how adding the d variable can control and improve segment shapes, getting more regular outlines with less disconnected subsegments, especially when it is compared from 16th segment onwards in this figure with the same range of segments in figures 3(a) and 4. The d variable was optimized using the NS or TNMU obtained when the r variable was computed.

It is possible to reduce the value of d with the purpose of getting even more regular shapes. This is done by relaxing the stop value using the percentage variable of equation (19). The OTNMU segmentation in figure 5(b) has $d \leqslant 5$, relaxing the TNMU by 10% as maximum,

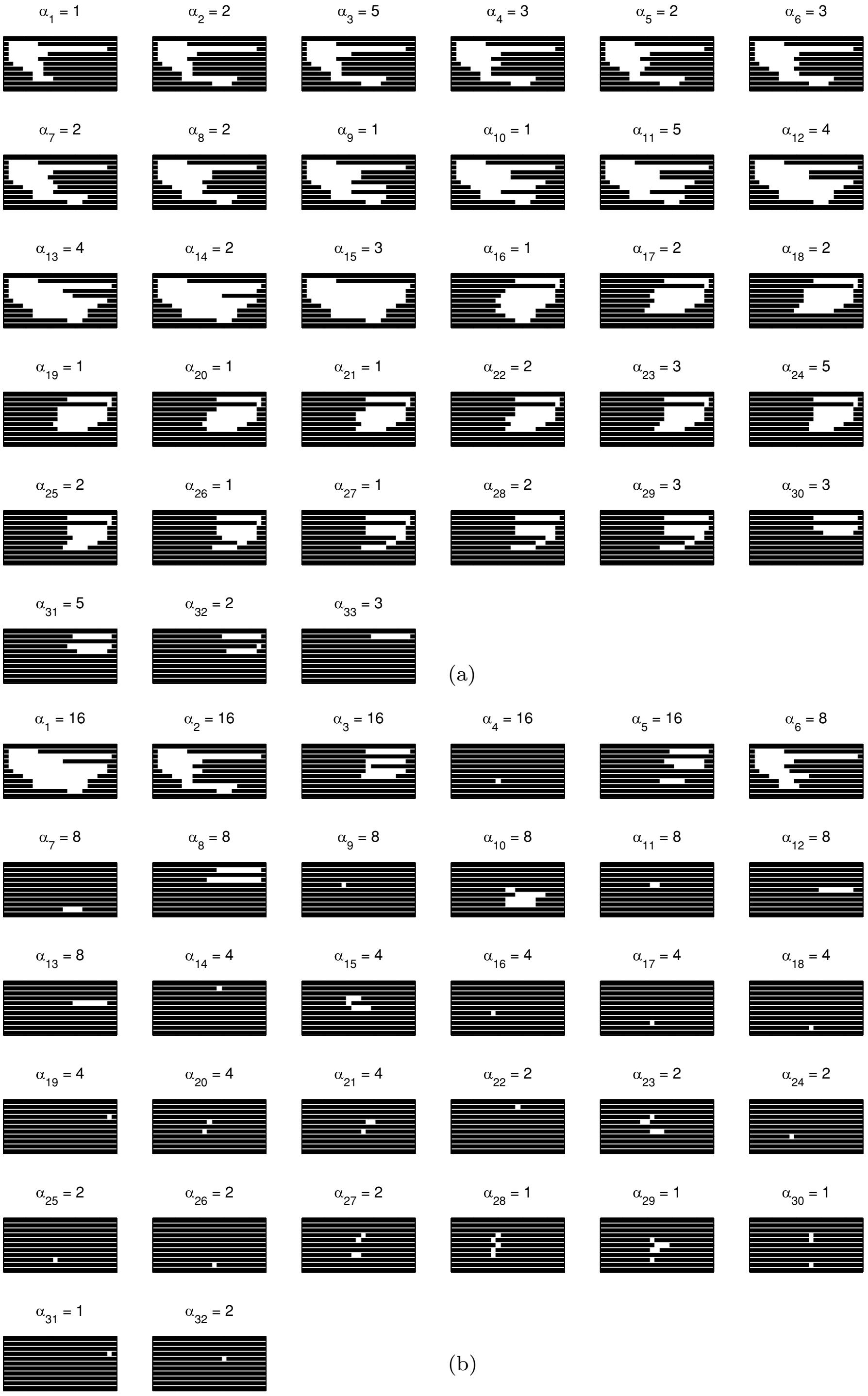


Figure 3. Segmentation of the 180° beam fluence matrix using RP and Xia and Verhey (1998) algorithms. (a) RP segmentation with 33 segments and 80 MU. (b) Xia and Verhey (1998) segmentation with 32 segments and 194 MU.

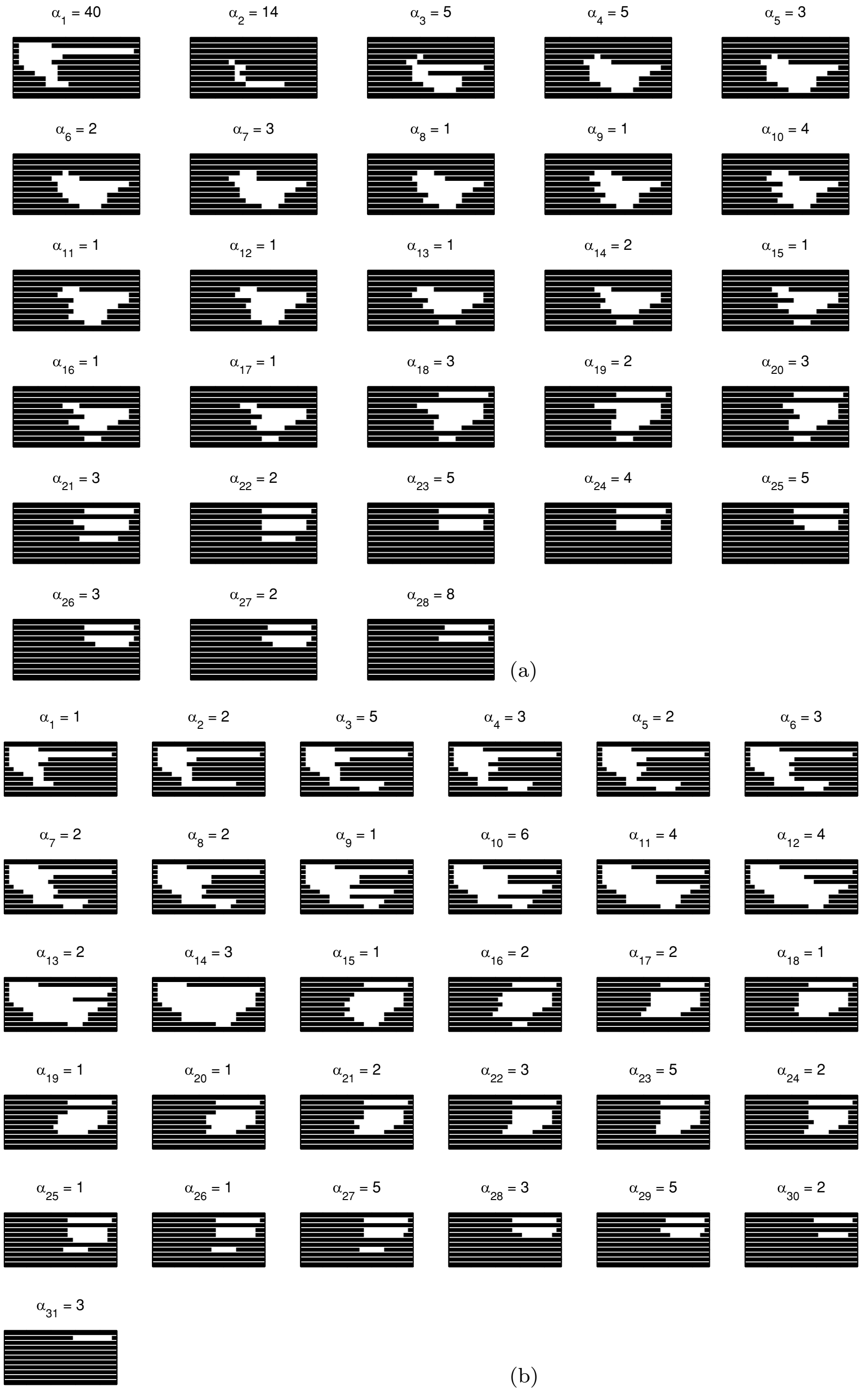


Figure 4. OTNMU and ONS segmentations not using segment shape control for the 180° beam fluence matrix. (a) ONS segmentation with 28 segments and 126 MU when $r = 23$. (b) OTNMU segmentation with 31 segments and 80 MU when $r = 23$.

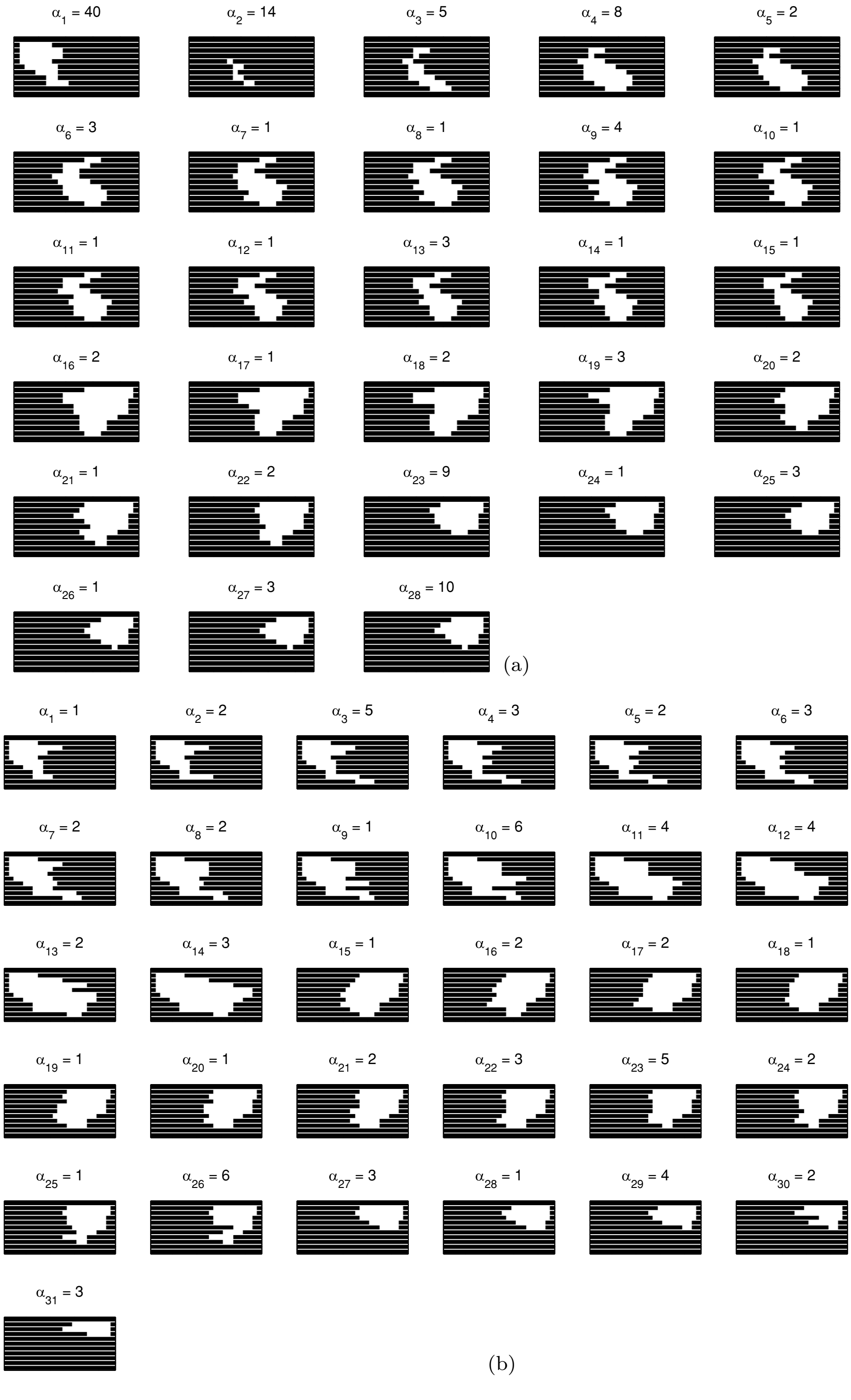


Figure 5. OTNMU and ONS segmentations using segment shape control for the 180° beam fluence matrix. (a) ONS segmentation with 28 segments and 126 MU when $r = 23$ and $d \leqslant 2$. (b) OTNMU segmentation with 31 segments and 80 MU when $r = 23$ and $d \leqslant 5$.

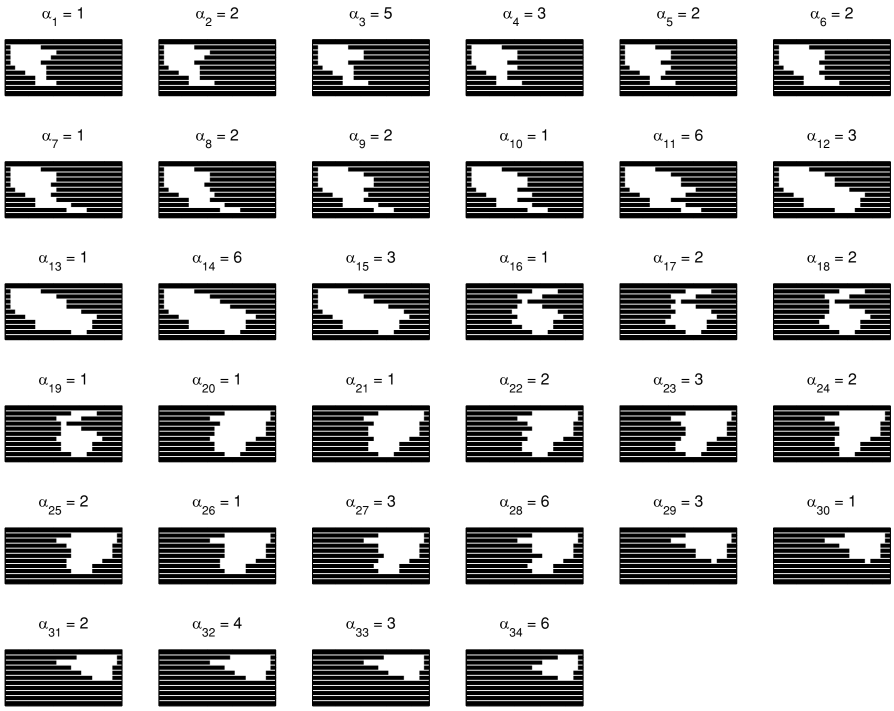


Figure 6. OTNMU segmentation with 34 segments and 86 MU when the TNMU is relaxed to reduce d, obtaining $r = 23$ and $d \leqslant 3$.

the result is a segmentation with $d \leqslant 3$ at the cost of 6 MU (7% more than the original). If a segmentation with $d \leqslant 2$ or lower is desired, the percentage could be increased. See figure 6 and compare with figure 5(b) the second and fifth rows, where the differences are more evident.

Finally, figure 7 illustrates the influence of the TGC on the segment shapes. The effect is similar to the one seen in figure 5 using the proposed segment shape constraint.

4. Discussion

Analysing the random test results in tables 2 and 3, it can be concluded that the proposed algorithms show good behaviour in terms of NS and TNMU, when comparing with other unidirectional segmentation methods, such as Bortfeld *et al* (1994) and the RP. In particular, the OTNMU gives almost identical results as the RP, whereas the ONS is slightly better than the RP and the OTNMU regarding the NS, but worst than both with respect to the TNMU, as one would expect.

The same behaviour can be observed in table 4 when looking at the real case test regarding both criteria for the RP, the OTNMU and the ONS, whereas the Xia and Verhey (1998) results in terms of the NS are not as good as those seen in the random test. However, the most remarkable difference is the improvement in the segment shapes observed in figures 5 and 6 compared to figures 3 and 4. The segment shape constraint has the properties of reducing the severe and intricate blocking, smoothing segment outlines and compacting segment shapes.

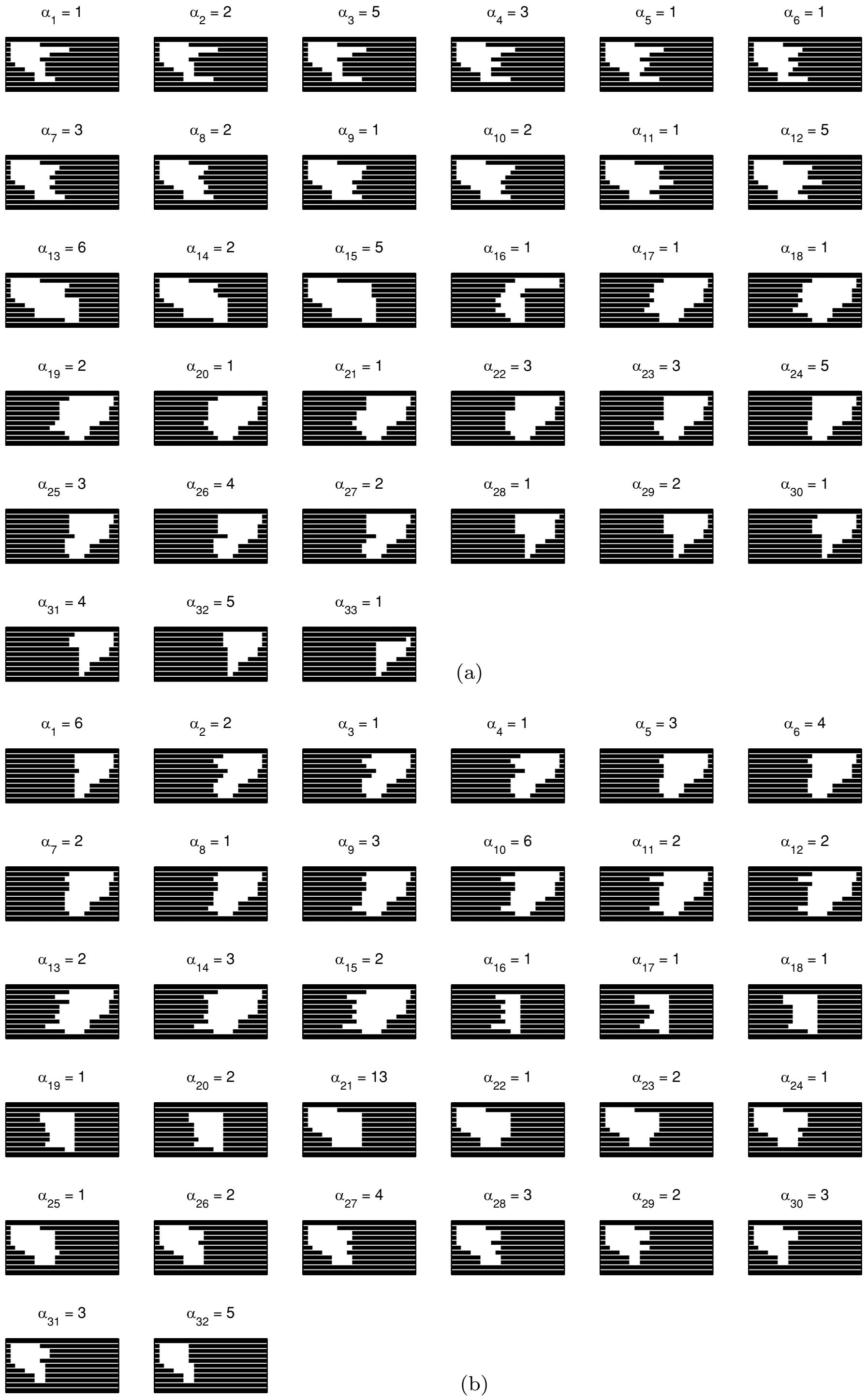


Figure 7. Example of the TGC applied on the OTNMU and RP algorithms, showing its influence on the segment shape. (a) OTNMU segmentation with 33 segments and 81 MU. (b) RP segmentation with 32 segments and 86 MU. The segmentation was from right to left.

Those properties come from the fact that limiting the difference between a right leaf and its adjacent leaves: (a) synchronizes their motion, creating a smoother segment 'front' (and outline), (b) contributes to the reduction of disconnected subsegments, minimizing the number of interleaved closed rows and (c) unidirectionality plus leaf motion synchronization will tend to avoid situations where the shape has a short Y-axis and a long X-axis (e.g. only two opposite leaves opened). The motion of one leaf depends on its adjacent leaves; one leaf (or a small set of leaves) could not advance much more than their neighbours. Thus, the segment will be more compact and regular as can be seen when comparing the latest segments in figures 3(a), 4 and 5.

The leaf synchronization achieved in figure 5 is obtained without increasing the NS or the TNMU. However, if it is still not enough, there is the possibility of relaxing the stop value for the NS or the TNMU, e.g. 5% more NS or MU, and get even better shapes, but it is not worth increasing it too much. Otherwise, segment shapes may be 'perfect' but the segmentation will not be feasible in practice, due to the increase in delivery time. There is a trade-off between both factors, shape versus NS and TNMU.

Finally, similar results were obtained in terms of shape when using the TGC, as can be seen comparing figures 7 and 5. However, table 4 shows that this constraint increases the NS and TNMU more than 10% and 8%, respectively, on each algorithm. In addition, the TGC influence on the NS or TNMU is quite unpredictable and cannot be limited or controlled.

5. Conclusions

A new MLC segmentation algorithm has been developed. For the computation of segments, a novel constraint for controlling the segment shapes has been added and two basic restrictions are considered, unidirectionality and the interleaf collision. The segment shape control will generate compact and regular shapes, unidirectionality will minimize leaf movements, thus reducing one treatment time factor, and the interleaf collision constraint will make suitable the proposed algorithm for MLCs with motion constraints.

For the computation of segment weights, the algorithm offers the possibility of selecting between two different criteria. The first criterion is a novel one proposed in this paper for minimizing the NS, and the second one is taken from Kalinowski (2006) and Engel (2005) for obtaining the optimal TNMU.

The results show that the algorithm works well compared to other published algorithms, having the bonus of shape control plus the criteria selection. Moreover, it is being transferred to the PCRT 3D® commercial software.

Acknowledgments

This work was supported by the Investigation Program of Aragón (Spain) Government, Project PM-021/2007. The authors would like to thank the company Técnicas Radiofísicas, S.L. (Spain) for allowing the use of the PCRT3D® software for the optimization plan obtained, and the anonymous referees for their valuable hints and suggestions.

Appendix A. Segmentation algorithm pseudocode

The following pseudocodes correspond to the main steps of the segmentation process:

Algorithm 1. Main function. Optimization of r and d variables.

```
function SEGMENTATION($A, criterion, percentage$): returns $S, \alpha$
    $r \leftarrow N; d \leftarrow N; collision \leftarrow true; max \leftarrow 0$          ▷ Subtraction optimization
    while $r \geqslant 0 \wedge collision$ do
        $< collision, auxS, aux\alpha > \leftarrow decomposition(A, r, d, criterion, max)$
        if $collision$ then
            $r \leftarrow r - 1$
        end if
    end while
    $d \leftarrow 0; collision \leftarrow false$          ▷ Depth optimization
    $max \leftarrow compute\ Maximum(S, \alpha, criterion, percentage)$
    while $d \leqslant N \wedge \neg collision$ do
        $< collision, auxS, aux\alpha > \leftarrow decomposition(A, r, d, criterion, max)$
        if $\neg collision$ then
            $S \leftarrow auxS; \alpha \leftarrow aux\alpha$
        else
            $d \leftarrow d + 1$
        end if
    end while
end function
```

Algorithm 2. Function computing the segmentation of a fluence matrix with given parameters. A, S and α are vectors containing the results for each iteration.

```
function DECOMPOSITION($F, r, d, criterion, max$): returns $collision, S, \alpha$
      $k \leftarrow 1; A(k) \leftarrow F; collision \leftarrow false$
      $lrPos \leftarrow initializeLeafPositions(A(k))$
      while $A(k) \rangle 0 \wedge \neg collision$ do
          $< collision, S(k) > \leftarrow computeSegment(A(k), r, d, lrPos)$
          if $\neg collision$ then
              switch $criterion$
                case $NS$
                  $\alpha_k \leftarrow computeNSWeight(A(k), S(k))$
                case $TNMU$
                  $\alpha_k \leftarrow computeTNMUWeight(A(k), S(k))$
            end switch
            $A(k+1) \leftarrow A(k) - \alpha(k) \cdot S(k)$
            if $max \neq 0$ then          ▷ Checking stop condition
                switch $criterion$          ▷ Collision boolean is used to stop, if necesary
                case $NS$
                    $collision \leftarrow (k > max)$          ▷ Checking max. NS
                case $TNMU$
                    $collison \leftarrow (sum(\alpha) > max)$          ▷ Checking max. TNMU
                end switch
            end if
            $k \leftarrow k + 1$
         end if
   end while
end function
```

References

Artacho J, Nasarre M, Bernues E and Cruz S 2007 A feasible application of constrained optimization in the IMRT system *IEEE T Bio.-Med. Eng.* **54** 370–9

Azcona J D, Siochi R A C and Azinovic I 2002 Quality assurance in IMRT: importance of the transmission through the jaws for an accurate calculation of absolute doses and relative distributions *Med. Phys.* **29** 269–74

Bortfeld T R, Kahler D L, Waldron T J and Boyer A L 1994 X-ray field compensation with multileaf collimators *Int. J. Radiat. Oncol. Biol. Phys.* **28** 723–30

Budgell G J, Mott J H, Williams P C and Brown K J 2000 Requirements for leaf position accuracy for dynamic multileaf collimation *Phys. Med. Biol.* **45** 1211–27

Cho P S and Marks R J 2000 Hardware-sensitive optimization for intensity modulated radiotherapy *Phys. Med. Biol.* **45** 429–40

Convery D J and Webb S 1998 Generation of discrete beam-intensity modulation by dynamic multileaf collimation under minimum leaf separation constraints *Phys. Med. Biol.* **43** 2521–38

Crooks S M, McAven L F, Robinson D F and Xing L 2002 Minimizing delivery time and monitor units in static IMRT by leaf-sequencing *Phys. Med. Biol.* **47** 3105–16

Dai J and Zhu Y 2001 Minimizing the number of segments in a delivery sequence for intensity-modulated radiation therapy with a multileaf collimator *Med. Phys.* **28** 2113–20

Dirkx M L, Heijmen B J and van Santvoort J P 1998 Leaf trajectory calculation for dynamic multileaf collimation to realize optimized fluence profiles *Phys. Med. Biol.* **43** 1171–84

Engel K 2005 A new algorithm for optimal multileaf collimator field segmentation *Discrete Appl. Math.* **152** 35–51

Galvin J M, Chen X G and Smith R M 1993 Combining multileaf fields to modulate fluence distributions *Int. J. Radiat. Oncol. Biol. Phys.* **27** 697–705

Hansen V N, Evans P M, Budgell G J, Mott J H, Williams P C, Brugmans M J, Wittkmper F W, Mijnheer B J and Brown K 1998 Quality assurance of the dose delivered by small radiation segments *Phys. Med. Biol.* **43** 2665–75

Hardemark B, Liander A, Rehbinder H and Löf J 2003 *Direct Machine Parameter Optimization with RayMachine in Pinnacle* Ray-Search Laboratories

Kalinowski T 2004 Multileaf collimator field segmentation without tongue-and-groove effect *Institut für* Mathematik, Universitt Rostock

Kalinowski T 2006 Realization of intensity modulated radiation fields using multileaf collimators *General Theory of Information Transfer and Combinatorics* (Berlin: Springer) pp 1010–55

Lorente J, Artacho J, Bernues E and Nasarre M 2006 Determination of optimal beam positions for conformal radiotherapy *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP 2006)* ed J Artacho **vol 2** pp II–II

LoSasso T, Chui C S and Ling C C 1998 Physical and dosimetric aspects of a multileaf collimation system used in the dynamic mode for implementing intensity modulated radiotherapy *Med. Phys.* **25** 1919–27

Memorial Sloan-Kettering Cancer Center. Departments of Medical Physics and Radiology R O 2003 *A Practical Guide to Intensity-Modulated Radiation Therapy* 1st edn (Medical Physics Publishing)

Que W 1999 Comparison of algorithms for multileaf collimator field segmentation *Med. Phys.* **26** 2390–6

Seco J, Evans P M and Webb S 2001 Analysis of the effects of the delivery technique on an IMRT plan: comparison for multiple static field, dynamic and nomos mimic collimation *Phys. Med. Biol.* **46** 3073–87

Shepard D M, Earl M A, Li X A, Naqvi S and Yu C 2002 Direct aperture optimization: a turnkey solution for step-and-shoot IMRT *Med. Phys.* **29** 1007–18

Siochi R A 1999 Minimizing static intensity modulation delivery time using an intensity solid paradigm *Int. J. Radiat. Oncol. Biol. Phys.* **43** 671–80

van Santvoort J P and Heijmen B J 1996 Dynamic multileaf collimation without 'tongue-and-groove' underdosage effects *Phys. Med. Biol.* **41** 2091–105

Xia P and Verhey L J 1998 Multileaf collimator leaf sequencing algorithm for intensity modulated beams with multiple static segments. *Med. Phys.* **25** 1424–34

Yang Y and Xing L 2003 Incorporating leaf transmission and head scatter corrections into step-and-shoot leaf sequences for IMRT *Int. J. Radiat. Oncol. Biol. Phys.* **55** 1121–34