

Fast processing of CBCT to improve delivered dose assessment

Kiran Joshi¹ and Tom Marchant¹

Christie Medical Physics and Engineering, The Christie NHS Foundation Trust,
Manchester, M20 4BX, UK kiran.joshi@physics.cr.man.ac.uk

Abstract. In a previous work we developed an algorithm to improve the Hounsfield Unit accuracy of CBCT images based on prior information from a CT scan. However, the processing time required to run the algorithm may be a barrier to clinical implementation. Here we describe work to speed up two key processing steps: 3D binary morphological operations and image interpolation.

Efficient binary morphological operators have been implemented in three dimensions using C++, extending the open source `leptonica` image processing library. Processing time comparisons have been made to implementations of three dimensional binary morphology available in ITK, MATLAB and IDL. The efficient implementations presented in this report have been found to require processing times up to three orders of magnitude shorter than the alternatives.

Image downsampling has also been investigated as a method to enable faster processing. Downsampling images by a factor of 2.0 (4.0) can produce a speedup of 1.8x (3.4x) in a processing step that interpolates into masked regions of an image. Preliminary studies of the effect of downsampling on the quality of final processed images suggest that downsampling by a factor of 2.0 produces a negligible decrease in final processed image quality. When used together these two developments can allow significantly faster processing of CBCT images.

1 Introduction

In recent years Cone Beam CT (CBCT) imaging systems have been integrated in the majority of linear accelerators used for radiation therapy. The kilovoltage (kV) x-ray source results in images with better soft tissue detail than those produced by megavoltage (MV) portal imaging or cone beam imaging systems. The kV-CBCT images are important for ensuring that patients are set up accurately prior to treatment. Since the images are acquired multiple times over the course of a treatment they also allow the possibility for adaptive radiotherapy. Delivered dose could be computed using the CBCT images, to assess whether the planned treatment remained optimal as tumour and patient reacted. It is often necessary, however, to apply image processing techniques to the CBCT image volumes, to restore accurate calibration before use in radiation therapy dose assessment. However given their large size – up to tens of millions of voxels per

image volume – even conceptually simple image processing operations such as binary erosion and dilation can be slow. Slightly more complex operations such as smoothing with a Gaussian kernel or interpolating into masked regions of an image can increase processing time considerably. The aim of the work presented here was (i) to produce an efficient implementation of three dimensional binary morphological operators and (ii) investigate the use of image downsampling, to allow CBCT image volumes to be processed as quickly as possible.

2 CBCT and Adaptive Radiotherapy

If significant changes in patient anatomy are observed in the CBCT images, current clinical practice often requires the acquisition of an additional fan-beam CT image. This enables the patient’s treatment to be re-planned to take into account the altered anatomy, but is an expensive and time-consuming process and results in the patient receiving a larger dose of radiation.

A preferable solution would be to re-plan the treatment directly using the CBCT image. However, the wide cone-beam of x-rays results in larger amounts of scatter than in conventional fan-beam CT scans, and produces artefacts that can make identification of anatomical structures difficult. The CBCT image voxels must also be corrected so that their values accurately represent x-ray attenuation. Dose calculations based on uncorrected CBCT images are prone to error, and can result in large dose inaccuracies when compared to doses calculated using fan-beam CT scans.

Several techniques have been proposed that aim to correct the CBCT images. One such correction procedure, proposed in [1], uses the information contained in the CT scan acquired prior to treatment to correct the overall normalization of the voxel values in the CBCT image, producing an image calibrated in Hounsfield units. The uniformity of the voxel values is also greatly improved by the correction procedure¹. In order to get maximum value out of the information contained in the CT images and to avoid degrading the quality of the CBCT images, the correction algorithm is applied to the large three-dimensional (3D) image volumes wherever possible. The resulting high quality CBCT image aides clinicians with tissue visualisation, and allows for the assessment of the delivered dose. In cases where large anatomical changes occur between CT and CBCT scans, this procedure is expected to be more robust than techniques such as deformable image registration. It is important that such a correction procedure can be performed quickly. If it is to have a large impact on clinical workflow the aim should be to correct the image volumes in as close to real-time as possible.

At various stages of the correction procedure 3D binary masks are created to select anatomical regions of the image (i.e. bone, soft tissue, gas). Fig. 1(a) shows an example of one slice of the image produced in a CT scan of a male pelvis. In Fig. 1(b) a threshold has been applied to separate the soft tissue (grey areas of

¹ An artefact present in CBCT images results in the voxels representing, for example, soft tissue such as muscle or fat having different values depending on their location in the image.

the mask) from the regions of bone and gas (black areas). Column (c) shows the binary mask after it has been eroded in three dimensions using a structuring element of size $5 \times 5 \times 3$ voxels. Binary erosion can be performed to prevent artefacts arising at the boundaries between the soft tissue and bone or gas when performing subsequent image processing techniques. The implementations of three

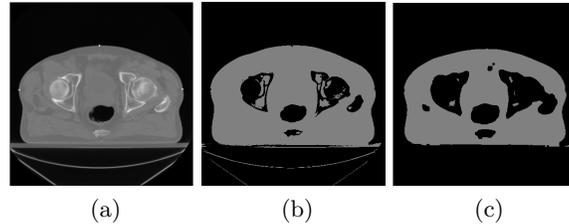


Fig. 1: (a): One slice of the image volume produced in a CT scan of a male pelvis. (b): A binary mask created after applying a threshold to separate soft tissue. (c): The result of eroding the binary mask in (b) using a 3D structuring element of size $5 \times 5 \times 3$.

dimensional binary morphology currently available in the Insight Toolkit (ITK) have been found to require processing times upwards of 1s (cf. Sect. 3.2). Given that binary morphological operations are performed many times over the course of the image correction procedure, the total time spent on binary morphology can become several seconds. Therefore, given the nature of the algorithm and the aim that images be processed in as little time as possible, work was undertaken on producing a more efficient implementation of three-dimensional binary morphology and is discussed in Sect. 3.

In a subsequent step of the correction algorithm, the masked regions of the image are interpolated into based on data from surrounding regions (i.e. image inpainting). This process is computationally intensive and must be performed for multiple masks. The interpolation of one image volume can take several seconds, depending on the size of the image. In an attempt to reduce the time taken to perform the interpolation downsampling was used. The amount of downsampling and the effect on the final corrected images is described in Sect. 4.

3 Efficient 3D binary morphology

3.1 Implementation

The three-dimensional binary morphological operations are implemented as an extension to the open-source image processing library `leptonica` [2], which provides efficient implementations of binary morphology in two dimensions. Full details of the two-dimensional implementations can be found in [3], though a

brief summary will be repeated here to aid in the discussion of the extension to three dimensions.

The basic binary morphological operations of dilation and erosion can be expressed in several ways. A standard definition given here lends itself well to demonstrating the low-level implementation. Let A represent an N-dimensional binary image, and B represent an N-dimensional binary structuring element (**Se1**). The dilation, \oplus , and erosion, \ominus , of an image A by **Se1** B are given, respectively, by

$$A \oplus B = \bigcup_{\mathbf{b} \in B} A_{\mathbf{b}} \quad (1)$$

$$A \ominus B = \bigcap_{\mathbf{b} \in \bar{B}} A_{\mathbf{b}} \quad (2)$$

where $A_{\mathbf{b}}$ is the image A after translating it by pixel vector \mathbf{b} . If the **Se1** is considered as a set of pixels with locations $b = (i, j)$ relative to an origin, typically at its centre, then the pixel vector \mathbf{b} is the vector pointing from the origin to pixel b . \bar{B} represents the inversion of B about its origin, i.e. $\bar{B} = \{-b \mid b \in B\}$.

In order to increase CPU and memory efficiency the binary images are packed such that each horizontal 32-pixel line segment is represented by a single 32-bit integer. I.e., each bit of the integer represents a single on/off pixel of the binary image. In this way the union and intersection operations can be implemented using bitwise OR and AND operators, and the required image translations can be implemented using the bit-shift operators available in C++. After packing the binary image each application of the bit-shift and bitwise logical operators can perform the necessary translate and logical OR/AND operations on 32 pixels at once.

Within the **leptonica** framework a packed two-dimensional image is represented by a **Pix** object. In the extension to three dimensions, each slice of the image is packed individually and the image volume is stored as an array of **Pix**. The object representing the structuring element must also be extended to take into account an additional dimension. Shifts in the z-direction are performed by selecting the corresponding slices of the volume from the array of **Pix**. The 3D morphological operations are then performed by looping over the required shifts defined by the **Se1**, selecting the necessary slices from the array of **Pix**, and taking the logical OR/AND of all corresponding pairs of 32-bit integers.

Boundary conditions are handled by padding the input image by an appropriate amount in each dimension prior to the application of the morphological operations. This padding must be performed by the user and is not implemented internally in the algorithm presented here².

The morphological operations presented here have been implemented in such a way that arbitrary-shaped structuring elements can be used. When the **Se1** is separable the morphological operations can be performed in the z-direction in-

² Padding must also be performed by the user when using the ITK, MATLAB and IDL binary morphology functions.

dependently, before applying two-dimensional morphological operations to each slice of the volume.

3.2 Performance

The performance of the algorithm implemented here, referred to as “Fast” in the following, is compared to the implementations of three dimensional binary morphology available in ITK³ (`itkBinaryDilateImageFilter`), IDL⁴ (`DILATE`) and MATLAB⁵ (`imdilate`). A binary mask is created from a CT scan of a male pelvis and dilated using separable, cube-shaped structuring elements with a range of sizes. The CPU time taken to perform the dilation is isolated from that taken to create the test image volume. For a given structuring element size the test is repeated multiple times to obtain an average dilation time. All tests are performed using a single core of a 3.33 GHz Intel Core i7 CPU in a machine with 16 GB of RAM.

Figure 2 shows the average times taken to dilate the image volume using cube-shaped structuring elements of various sizes. CPU times taken to perform a binary erosion show the same trends as those shown in Figure 2.

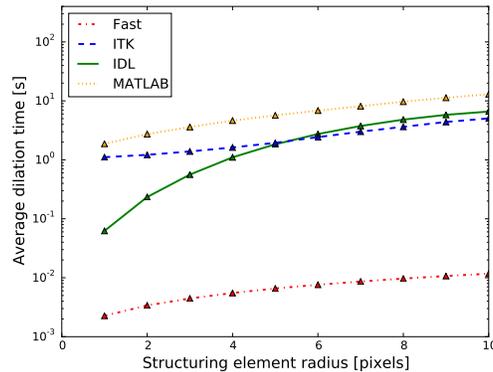


Fig. 2: Average CPU times spent performing binary dilation on a 512x512x31 voxel binary mask created from a CT scan, using structuring elements of various sizes. The structuring element radius is the distance, in voxels, from the centre of the structuring element to the outer faces. A `Se1` of radius 1 is therefore a 3x3x3 cube. The fast implementation presented in this report (red, dot-dashed) is compared to the implementations available in ITK (blue, dashed), IDL (green, solid) and MATLAB (orange, dotted). The fast implementation is typically two to three orders of magnitude faster than the alternatives.

³ ITK v.4.7.2, The Insight Software Consortium

⁴ IDL v7.1, Exelis Visual Information Studio, Boulder, Colorado, USA

⁵ MATLAB v6.5 Release 13, The Mathworks, Inc., Natick, Massachusetts, USA

Despite the ITK implementation making use of the techniques presented in [4] it is only able to perform binary morphological operations in times longer than 1s.

The IDL implementation runs in times an order of magnitude shorter than ITK when using very small structuring elements. However the processing time increases rapidly as the `Se1` radius increases.

Binary dilation implemented in MATLAB has approximately the same performance as ITK, with average processing times varying between 1.8s and 14.7s.

The fast binary morphological operators implemented in this report typically run several hundreds of times faster than the functions available in ITK, IDL and MATLAB. When dilating the CT mask using very small `Se1`s the processing time required is just 0.002s, compared to 0.06s with IDL and 1.1s with ITK.

The resulting dilated masks produced by each of the algorithms were compared by subtraction, and were found to be identical.

4 Image downsampling

Interpolation into masked regions of the image volumes is performed on each slice of the image volume separately, often referred to as “2.5D”, and takes place prior to the creation of a correction map that is subsequently applied to the CBCT image. The temporary images used to create the correction map are downsampled using the ITK `ResampleImageFilter`, with linear interpolation used to compute pixel values at non-grid locations of the full-sized images. The original CBCT and CT images themselves are not downsampled at any point.

Since the interpolation is performed in 2.5D, the downsampling is only performed in the x- and y- directions; the number of slices in the volume is kept fixed. For example, a downsample factor of 2.0 implies that an image slice of size 410x410 pixels becomes 205x205 pixels. The computation time taken to perform the interpolation after different amounts of downsampling is shown in Table 1. Downsampling the images by a factor of 2.0 results in a speed increase of 1.8x,

Downsample factor	Mean time/image [s]
1.0	1.29
2.0	0.70
4.0	0.38

Table 1: Variation of 2.5D interpolation speed with downsampling factor.

and downsampling by a factor of 4.0 results in a 3.4x speedup.

The effect of the downsampling, and subsequent upsampling, on the final corrected CBCT images was estimated using two methods: by assessing the uniformity of the various CBCT images corrected after downsampling by different

amounts, and by subtracting the CBCT images corrected after downsampling from the CBCT image corrected without downsampling.

The image uniformity was assessed by defining small regions of muscle tissue which were expected to have similar densities. Examples of some of the regions are shown in Fig. 3(a)-(c). The same set of regions were used when analysing all images. The mean pixel value in each of the regions was calculated and the average and standard deviation of these means was used to quantify the uniformity. As well as the corrected CBCT images, the uniformity was calculated using a CT image of the same patient. A comparison of the uniformity in each of the images is show in Table 2.

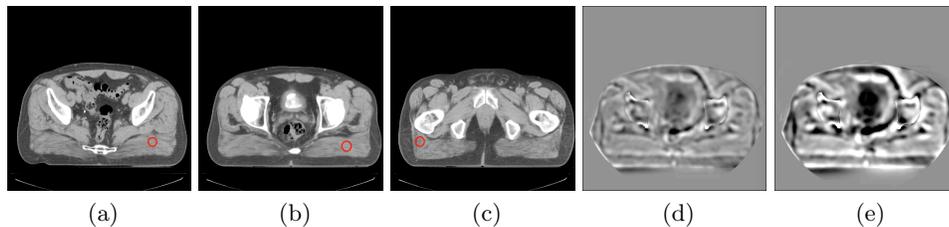


Fig. 3: (a)-(c): Small regions containing similar types of muscle tissue, in different parts of the image, were used to estimate image uniformity. (d), (e) CBCT images corrected using downsampling factors of 2.0 and 4.0, respectively subtracted from that corrected without downsampling.

Image	Average \pm std. dev.
CT	1011.8 \pm 16.3
Uncorrected CBCT	1044.2 \pm 47.8
Corrected CBCT	1010.0 \pm 16.6
Corrected CBCT, Downsample 2.0	1008.6 \pm 17.7
Corrected CBCT, Downsample 4.0	1006.9 \pm 18.5

Table 2: Comparisons of image uniformity estimates.

Little difference is observed between the uniformities of the corrected CBCT images. The average of the mean pixel values in the muscle tissue regions remains approximately constant, while the standard deviation (non-uniformity) increases slightly as the amount of downsampling increases.

The result of subtracting the CBCT image corrected using a downsampling factor of 2.0 (4.0) from the CBCT image corrected without downsampling is

shown in Fig. 3d (e). When downsampling by a factor of 2.0 the differences are typically small. Differences in pixel values of around 10 are observed across the majority of the image. In regions near the edges of bones, and around the outer edge of the patient, the differences are slightly larger with maximum differences of approximately 70 appearing in isolated areas. When downsampling by a factor of 4.0 the differences are increased. Across the majority of the image differences in pixel value are approximately 15-20, with maximum differences of around 150.

Whether differences of this size can be tolerated is unclear. The ultimate test of the effect of downsampling will be to use the various corrected CBCT images to calculate the dose delivered by a radiotherapy plan.

5 Summary

Two methods have been investigated to speed up processing of CBCT images for delivered dose assessment. The efficient implementation of 3D binary morphological operators has been presented and their performance compared with the implementations available in ITK, IDL and MATLAB. The algorithms are implemented as an extension to the `leptonica` image processing library and have been found to run up to three orders of magnitude more quickly than commonly available alternatives. It is likely that these fast 3D operators could be of use in a wide range of applications.

Image downsampling has also been investigated as a method for enabling faster image processing. Downsampling by a factor of 2.0 or 4.0 can significantly reduce overall processing times. The effect on the final corrected images has been estimated by directly comparing corrected images and assessing image uniformity. Preliminary results suggest that downsampling by a factor of 2.0 has a negligible impact on the quality of the corrected image, though further work will be performed by using corrected CBCT images to calculate dose delivered by a radiotherapy plan.

The combination of these techniques produces an overall speedup of 3x and allows CBCT images to be corrected in under 10s.

This research is funded by UK MRC grant MR/L023059/1.

References

- [1] T E Marchant et al. “Shading correction algorithm for improvement of cone-beam CT images in radiotherapy”. In: *Physics in Medicine and Biology* 53.20 (2008), p. 5719.
- [2] D S Bloomberg. URL: <http://www.leptonica.com>.
- [3] D S Bloomberg. “Implementation Efficiency of Binary Morphology”. In: *International Symposium for Mathematical Morphology VI* (2002). URL: <http://www.leptonica.com/papers/binmorph.pdf>.
- [4] N. Nikopoulos and I. Pitas. “A fast implementation of 3-D binary morphological transformations”. In: *Image Processing, IEEE Transactions on*, 9.2 (Feb. 2000), pp. 283–286. ISSN: 1057-7149. DOI: 10.1109/83.821743.