Efficient Morphological Reconstruction: A Downhill Filter

Kevin Robinson [*] Paul F. Whelan

Vision Systems Group, Dublin City University, Ireland

Abstract

The downhill filter is an elegant and efficient single pass reconstruction algorithm which demonstrates fast and consistent performance. It operates through a controlled process of region growing by ordered aggregation of surface pixels onto an expanding shell.

Key words: Reconstruction by Dilation, Downhill Filter, Segmentation, Mathematical Morphology

1 Introduction

The goal of the downhill filter is to suppress unwanted regions of high intensity signal while maintaining signal intensity in the regions of interest, which are seeded with markers in order to initialise the algorithm. The motivation for this work stems from an investigation into the segmentation of complex treelike structures in MRCP (Magnetic Resonance Cholangiopancreatography) data (Sai and Ariyama, 2000; Robinson et al., 2002). This relates to an area of medical image analysis which addresses the task of obtaining the optimum classification on a highly branched and weakly delineated structure in relatively noisy, low resolution data. Figure 1 shows a maximum intensity projection rendering of a volumetric ($298 \times 298 \times 60$ voxels) MRCP dataset, illustrating the segmentation task.

As part of this investigation we considered the use of reconstruction by dilation (Serra, 1982; Soille, 1999) as a means of attenuating unwanted portions of the signal. We quickly identified many and varied instances of the use of this technique (e.g. Angulo and Serra, 2003; Salembier et al., 1996; Metzler et al.,

[*] Corresponding author. *Email:* `kevin.robinson@eeng.dcu.ie`

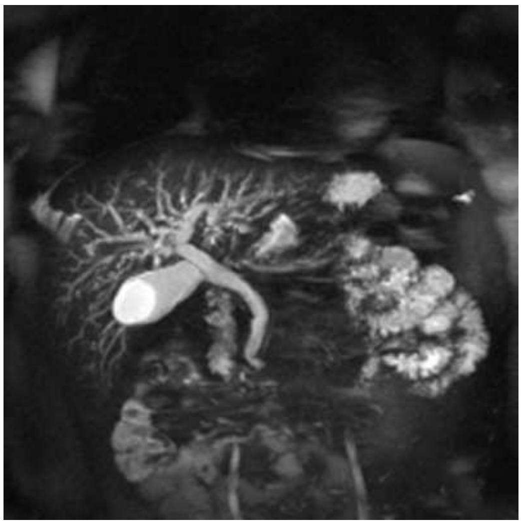


Fig. 1. Maximum intensity projection (MIP) rendering of an MRCP dataset showing a highly branched biliary tree and prominent, non-relevant neighbouring regions.

2001; Araujo et al., 2001), but also noted the inefficiency of the 'iterate until convergence' approach often used to implement it.

A number of optimisations and algorithmic efficiencies have been detailed for this and similar procedures in both binary and greyscale morphology. In particular Vincent (1993) examines efficient approaches to greyscale morphological reconstruction by dilation, while others have addressed closely related issues in efficient morphological filtering in general, including structuring element decomposition and manipulation (Park and Yoo, 2001; Sivakumar et al., 2000; van Droogenbroeck and Talbot, 1996), flat zones (Salembier and Serra, 1995), interval coding (Ji et al., 1989), and ordered queues (Vincent and Soille, 1991; van Vliet and Verwer, 1988; Beucher and Meyer, 1993).

These enhancements notwithstanding the procedure remains computationally expensive and highly data dependant. We have developed an approach which achieves the desired filtering in a single pass through the data and as such is both fast and linear time in its execution.

1.1 Morphological Reconstruction by Dilation

We start with a concise review of the basic principles and definitions behind reconstruction by dilation, showing how it relates to the standard and conditional dilation operators, and is itself defined in terms of iterated geodesic dilations.

We can represent a dilation of size N by $\delta^{(N)}$, as in Eq. 1, where I is the image, p and q are pixels, D represents the image domain, and N_G is the

Table 1
Key to mathematical notation.

Symbol		Definition
$\delta^{(N)}I$	–	Dilation of size N, applied to image I
$\delta_C^{\star(N)}I$	–	Conditional dilation of size N, applied to I, conditioned on C
$\delta_C^{(N)}I$	–	Geodesic dilation of size N, applied to I, conditioned on C
$\forall a : B \bullet P(a)$	–	Universal quantification: for all a in B predicate $P(a)$ is true
$\{a : B \mid P(a)\}$	–	Set Comprehension: the set of all a in B that satisfy predicate $P(a)$
$\widehat{=}$	–	Is equal to by definition
$\leftarrow$	–	Assignment (to differentiate from = for equality)
$\Rightarrow$	–	Implication
$\frown$	–	List catenation
head	–	Returns the first element in a list
tail	–	Returns a list minus its head

pixel neighbourhood addressed in the dilation. The mathematical notation used here and throughout this paper follows the formal software engineering specification style described in Woodcock and Loomes (1988). A concise key to the less familiar aspects of the notation used is provided in Table 1.

$$\delta^{(N)}I \;\widehat{=}\; \forall p : D \bullet \forall q : N_G(p) \bullet (I[q] \leftarrow max(I[p], I[q])) \tag{1}$$

Eqs. 2 and 3 represent respectively conditional and geodesic dilation of size N, where I is the marker, C is the mask, and $\wedge$ is the point-wise minimum operator. Reconstruction by dilation is defined as the geodesic dilation operator, $\delta_C^{(1)}$ applied iteratively until stability.

$$\delta_C^{\star(N)}I \;\widehat{=}\; \delta^{(N)}I \wedge C \tag{2}$$

$$\delta_C^{(N)}I \;\widehat{=}\; \delta^{(1)}I \wedge C \ldots N\ times \tag{3}$$

In Figure 2 the operation of dilation, conditional dilation, and geodesic dilation on greyscale data is illustrated. Figure 2c shows the dilation of the marker in 2b using a standard 1-D two connected structuring element. Conditional and geodesic dilations are shown in Figures 2d and 2e respectively. The top row illustrates a dilation of size five while in the bottom row the operators have been run to completion.

The plot in Figure 2e on the second row shows the geodesic dilation iterated until stability and as such is by definition the reconstruction by dilation of the signal in 2b conditioned on the signal in 2a. Study of the plot reveals the 'downhill' nature of the process which informed the approach taken in developing our algorithm and gives it its name of downhill filtering. Moving away from the maxima in the starting signal the filtered signal falls with the conditioning mask, but once it has fallen it does not rise again as the conditioning mask rises. This behaviour means that marked high intensity

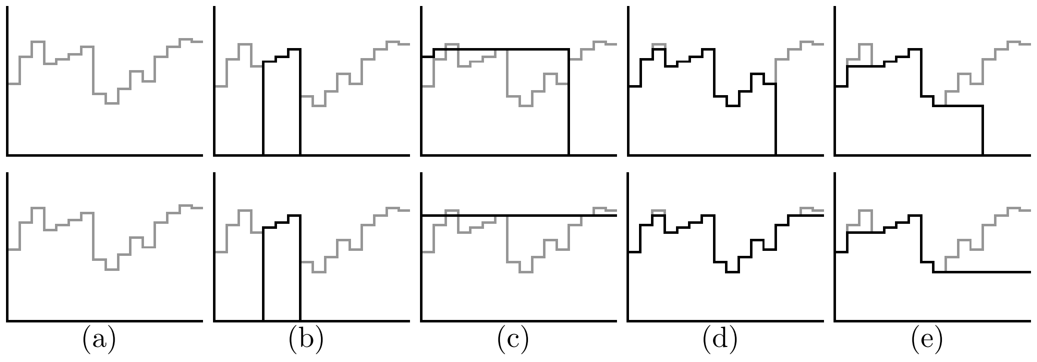


Fig. 2. A comparison of standard, conditional, and geodesic dilation, after five iterations (top row), and iterated until stability (bottom row); (a) the conditioning mask, (b) the starting signal, (c) standard dilation, (d) conditional dilation, (e) geodesic dilation. The conditioning mask is overlayed in grey onto all plots for clarity.

regions will be retained while those that are unmarked will be attenuated to the level of the highest valley separating them from any marked regions.

1.2 Existing Algorithms

In Vincent (1993) four successively more efficient algorithms are presented. The last of these, referred to below as algorithm D proved to be the most efficient of all the existing algorithms tested. In order to both confirm the operation and gauge the performance of our approach we implemented the four algorithms which Vincent (1993) describes, to test alongside our own. Detailed descriptions of these approaches can be found in that paper. Brief outlines are given here for reference:

A *Standard Technique:* directly follows the 'iterate to completion' scheme which defines reconstruction by dilation. Combined dilation and pointwise minimum operations are iterated until stability.

B *Sequential Reconstruction:* the number of iterations required is reduced significantly by alternating raster and anti-raster scannings of the data and allowing changes so far in the current iteration to be included in the calculations.

C *Reconstruction Using a Queue of Pixels:* a FIFO queue is initialised with boundary pixels from the regional maxima. Pixels are then removed, their neighbours examined, changed, and added to the queue as required. Processing continues until the queue is empty.

D *Hybrid Reconstruction Algorithm:* this combines features of the previous two. It initialises the FIFO queue during a single pass of the raster/anti-raster scheme of algorithm B. It then proceeds to process pixels in the same fashion as algorithm C until the queue is empty.

2 Downhill Filter

As with the third and fourth algorithms above the downhill filter operates on a pixel queue. However instead of a FIFO queue, a random access queue is implemented in order to allow the processing of pixels in an optimal order thus guaranteeing that every pixel is processed only once. This approach yields a filter which is extremely efficient, and perhaps more importantly is data insensitive in its execution time.

This is in contrast to the above four algorithms all of which are highly data dependant. Folded or rolled up structures in the input image for instance seriously compromise the execution speeds achieved by all four approaches so that no guarantees can be given as to the processing time required in the general case. Our algorithm exhibits no such level of variability in its execution speed.

The underlying principle behind the algorithm can be stated in the following terms. Given that it is known which pixels have so far been finalised it is possible to finalise the next pixels thus. Find the highest valued finalised pixel which has one or more non-finalised neighbours. Each non-finalised neighbour may now be finalised to be the lesser of: its mask value and the value of the aforementioned finalised pixel. This process continues until no non-finalised pixels remain. Effectively it can be thought of as having a shell of pixels around the finalised regions which expands in a controlled fashion, resulting in the desired reconstruction.

We present the development of the algorithm in two stages. In the first case we restrict the nature of the marker image provided as input to the filter. This simplifies the formulation of the algorithm and represents the form in which we originally developed it, reflecting the particular requirements of our application. This initial approach is to strengthen the usual precondition so that each pixel in the marker is either equal to the corresponding pixel in the mask or it is equal to zero, as shown in Eq. 4, where $\vee$ is the logical OR connective.

$$\forall p : D \bullet ((I[p] = C[p]) \vee (I[p] = 0)) \tag{4}$$

In the general formulation of reconstruction by dilation the precondition is presented in a weaker form (Eq. 5) so as to insist only that the marker image is pixel-wise less than or equal to the mask image.

$$\forall p : D \bullet (I[p] \leqslant C[p]) \tag{5}$$

Our strengthened form covers a very useful subset of the general reconstruction problem where seed regions in the marker are extracted directly from the mask image in order to initialise the reconstruction, and everything else in the marker is initialised to zero. Later we will relax the precondition to it's more general form and extend our algorithm correspondingly, allowing the initial marker to take any form less than or equal to the mask. This latter more general form is sometimes useful or necessary, for instance in the calculation of the h-domes of an image (Vincent, 1993).

2.1 The Strengthened Form

The reconstruction is determined directly in the marker image I. Initially each pixel in I is either equal to the corresponding pixel in the mask image C or it is equal to zero. Therefore by definition all non-zero pixels in I are at their final value. Let m be the maximum value of the pixels in I. We note that no pixel in the final image can have a value greater than m.

We will need m lists indicated $L[1]$ to $L[m]$, each corresponding to a greylevel between 1 and m. As a pixel is set to its final value it is placed in the list corresponding to that value. An examination of the efficient handling of the lists and the tracking of which pixels are finalised at any point is beyond the scope of this discussion. C code and test images are available for download on the Vision Systems Group Code Archive[1], which will clarify implementation details. Here we will proceed in general terms.

Initialisation consists of placing each non-zero pixel in I into the appropriate list. Processing then proceeds starting with list m and continuing down towards list *one*. While the current list is not empty the next element is removed and its neighbourhood is examined. For each neighbour which has not already been finalised, I is set equal to the lesser of the current list number and the value in C at this location, and the neighbour is added to the corresponding list and marked as finalised. This scheme achieves the ordered processing of pixels required by the algorithm and guarantees that the filtered result is grown out from the seeded regions, from high to low pixel intensity, such that the final image contains the reconstruction sought.

[1] *URL:* `http://www.eeng.dcu.ie/~vsl/vsgcode.html`

Algorithm: *reconstruction by downhill filtering (strengthened precondition):*

```
• Find $m$, the maximum pixel value in marker image $I$
    $\{m : \mathbb{N} \mid m \in I \wedge \forall p : D \bullet (m \geqslant I[p])\}$
• Place each non-zero pixel in $I$ into its appropriate list
    $\forall p : D \bullet (I[p] \neq 0 \Rightarrow L[I[p]] \leftarrow L[I[p]]^\frown p)$
• Process the $m$ lists from high to low:
    $For\ n = m..1$
      $While\ L[n] \neq \emptyset$
        $p \leftarrow head(L[n])$
        $L[n] \leftarrow tail(L[n])$
        $For\ q \in N_G(p)$
          $If\ C[q] > 0 \wedge !finalised(I[q])$
            $I[q] \leftarrow min(n, C[q])$
            $L[I[q]] \leftarrow L[I[q]]^\frown q$
```

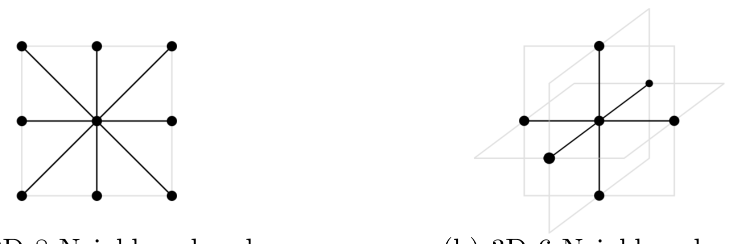
2.2 Relaxing the Precondition

If we now weaken the precondition on I it can no longer be stated that all nonzero pixels in I are at their final value to start with. We still initialise as before but we must now cater for the fact that we may need to remove pixels from one list (where they were initialised) and place them onto another list (where they are finalised). Since the lists are processed from high to low a pixel which needs moving will always be moved and finalised before its initial entry is ever reached. Thus it is still the case that no pixel is processed more than once.

As before the reconstruction is determined directly in the marker image I. Initially each pixel in I is less than or equal to C. Initialisation is as before and processing again proceeds starting with list m. While the current list is not empty the next element is removed and its neighbourhood is examined. For each neighbour which has not already been finalised, I is set equal to the lesser of the current list number and the value in C at this location, and the neighbour is added to the corresponding list. If it was already in a list it is also removed from that list.

3 Results

We implemented the five algorithms discussed and conducted a number of tests using a wide range of image and volume datasets. A two dimensional eight connected structuring element neighbourhood was used with each algorithm for tests conducted on the image data, Figure 3a, and a three dimensional six connected neighbourhood was used with the volumes, Figure 3b.



(a) 2D 8-Neighbourhood (b) 3D 6-Neighbourhood

Fig. 3. Structuring elements used in the processing of the 2D and 3D datasets.

3.1 Validation Tests

In assessing the algorithms we ran validation tests to confirm that all were both well formed and correctly implemented. The algorithmic validity of each implementation was checked against the mathematical definition of reconstruction by dilation, confirming that each approach reduces to the same fundamental operation.

In order to test that our implementations were error free each function was then applied to a range of images and volumes and we confirmed that all produced identical results. Algorithm A, which directly implements the definition of reconstruction by dilation using iterated geodesic dilations was taken as providing the baseline results for our comparisons. The output generated by each of the other algorithms was programmatically compared against that of algorithm A, pixel for pixel, and we thus confirmed that all algorithms performed correctly. Figure 4 shows the results achieved on three of the test datasets used.

In Figure 4a the seed was placed at the outermost end of the spiral arm in the top right hand portion of the image, and the four unseeded regions (the border, the unmarked spiral, and the two regions at the centre) were eliminated. In Figure 4b the seed was placed in the upper U-bend of the looped region. The three unconnected horizontal bars were thus removed in the filtered result. In Figure 4c the seed was placed in the broad descending trunk of the tree structure we wished to isolate and again the unconnected high intensity regions in the image were eliminated.

Figure 5 shows two volumetric datasets, rendered in maximum intensity projection, each before and after filtering. As the data is three dimensional some regions of high intensity signal which appear in the top two renderings to be in contact with the seeded region but which are in fact in different planes have

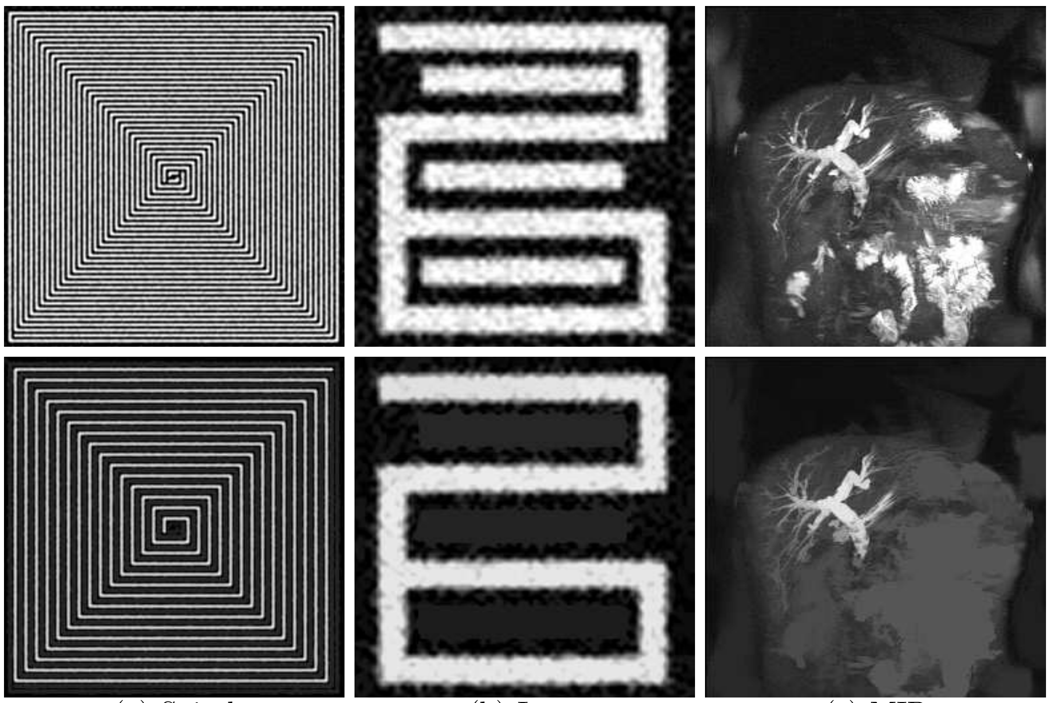

(a) Spiral (b) Loop (c) MIP

Fig. 4. Filtering results on three test images. The top row shows the original data while the bottom row shows the filtered results.

been eliminated from the filtered results, shown in the bottom two images. This illustrates how the filter can be used to clear the field around the objects of interest in a volume allowing subsequent processing to better focus on the true regions of interest.

3.2 Timing Tests

Once the basic validation procedures had been completed we moved on to conduct timing tests in order to characterise the relative performance of the five approaches on a range of data. Tables 2 and 3 document the results of these tests for five image and five volume datasets respectively. The methods are labelled alphabetically in correspondence with the labelling given in Section 1.2, with the downhill filter being added to the classification as method E. For comparison purposes we note that all tests were performed on a 1.8GHz Pentium 4 with 512MB of RAM.

Table 2 presents execution times for each of the five algorithms. As expected the standard technique, method A exhibits extremely poor performance relative to all the rest. Method E, the downhill filter demonstrates a clear superiority in execution times across a range of data, with only the blank image yielding better times in any of the other algorithms, due to the higher initiali-

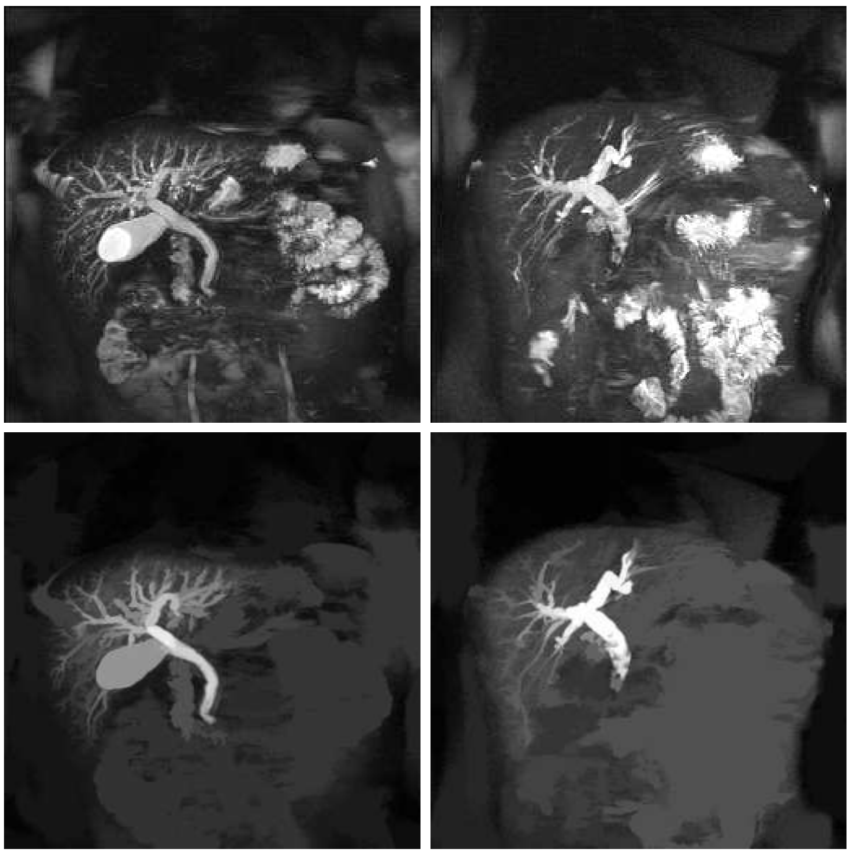


Fig. 5. Maximum intensity projections of two volume datasets.

Table 2
Timings in milliseconds for the five algorithms applied to five test images. Bracketed numbers indicate iterations required in methods A and B. Methods C, D, and E are queue based and as such do not yield an iteration count.

Method	Spiral		Loop		MIP 1		MIP 2		Blank	
A	(7191)	47898.0	(409)	2884.3	(270)	1892.1	(247)	1735.0	(130)	700.0
B	(17)	59.4	(8)	29.7	(9)	37.5	(7)	28.6	(2)	6.2
C		175.0		42.2		40.6		37.5		14.0
D		139.1		43.5		28.2		21.8		4.6
E		10.9		9.4		9.3		9.4		7.8

sation overheads involved in method E. In all cases the image size is 256×256 pixels.

Note also that the times for method E vary by only a couple of milliseconds across the entire range of test images (a variation of less than 40%) whilst the most consistent of the rest (method B) shows variations of close to a factor of ten in its execution times. The largest intra-method variation approaches a seventy times difference from best to worst.

This reflects the guaranteed single pass nature of the downhill filter in contrast to the effectively unbounded nature of the other approaches. We attribute the small variations which do exist in the execution times for method E to varying caching related overheads incurred on the hardware platform employed, based on the differing order in which pixels are addressed throughout the image from one dataset to the next.

The first test image consists primarily of two interlaced spirals and was intended most especially to stress methods A and B which both address the image pixels in scan order and as such are particularly prone to poor performance when faced with any kind of wrapped up structure in the input image. Methods C and D also fare particularly badly in this case due to the indiscriminate way in which pixels are added to the FIFO queue for processing, which can result in pixels being processed many times before their final value is reached. Method E by contrast demonstrates little variation in its execution time for this image as compared with the other test images illustrating its stability across all data configurations.

The next three images, the synthetic loop and the two maximum intensity projections, represent data in a more usual range of image structure, illustrating the typical performance to be expected of each of the tested methods. Once again method E performs most efficiently, while method A as expected fails to approach the performance of the other algorithms.

The final image is a constant mid-grey and represents the most trivial form of input. With a single pixel at the centre of the image used as a marker, method A still requires as many iterations as the most distant pixel in the image in order to reach completion. Methods B, C, and D all demonstrate their ability to take advantage of the simple configuration and all achieve low execution times, with B and D actually outperforming E for the first time. Method E itself continues to perform well with its single pass execution path once more in evidence maintaining a consistent performance.

Table 3 shows results for five test volumes. Here the variations are even more pronounced than with the image data, as once again timings on the standard technique, method A, illustrate how slow this method can be, while the downhill filter again achieves the best execution times across a broad range of datasets.

Again the five datasets used cover a number of levels of complexity. All volumes are of the same size ($298 \times 298 \times 60$ voxels). The first two are synthetic volumes consisting of interlaced spirals of different configurations designed to stress the algorithms. Method A takes almost thirty hours, and over a quarter of a million iterations to fully process the first of these. Method E does exactly the same job in just 2.5 seconds.

Volumes three and four were included to represent data within the usual expected input range. And as before a blank, constant mid-grey dataset was

Table 3
Timings in seconds for the five algorithms applied to five test volumes. Bracketed numbers indicate iterations required in methods A and B. Methods C, D, and E are queue based and as such do not yield an iteration count.

Method	3D Spiral		3D Roll		Study 1		Study 2		Blank	
A	(255003)	104868.9	(13643)	6051.1	(742)	403.0	(428)	227.5	(370)	137.0
B	(542)	189.0	(20)	7.5	(17)	8.9	(24)	12.5	(2)	0.7
C		58.4		53.7		23.6		21.5		2.6
D		54.2		54.1		14.7		12.6		0.5
E		2.5		1.9		3.4		3.3		1.2

used to test the most trivial case of input data, resulting again in methods B and D outperforming the downhill filter for the only time.

4 Conclusion

The downhill filter offers an efficient alternative to existing reconstruction by dilation algorithms. It exhibits a number of clear advantages over the other approaches examined. The most important of these may be summarised as follows:

1 *Short execution time:* The downhill filter proved to be faster than any of the other algorithms tested, in all but the most trivial of cases where the algorithmic overheads dominate.
2 *Consistent execution time:* Unlike the other algorithms tested execution times are not dependant on the nature of the data processed. This makes the downhill filters behaviour predictable as well as fast.
3 *Simple formulation:* The concise nature of the downhill filter algorithm leads to a clear and compact formulation. Our basic implementation occupies just forty three lines of C code.

This paper demonstrates the downhill filters excellent performance and in that context shows reconstruction by dilation to be an attractive and highly usable filtering tool ideal for many image processing tasks and well suited to use in time sensitive, real time applications.

References

Angulo, J., Serra, J., 2003. Automatic analysis of dna microarray images using mathematical morphology. Bioinformatics 19 (5), 553–562.

Araujo, A., Guimaraes, S., Cerqueira, G., 2001. A new approach for old movie restoration. In: Proc. SPIE-High-speed Imaging and Sequence Analysis. pp. 67–77.

Beucher, S., Meyer, F., 1993. The morphological approach to segmentation: The watershed transformation. In: Dougherty, E. (Ed.), Mathematical Morphology in Image Processing. Marcel Dekker Inc, pp. 433–481.

Ji, L., Piper, J., Tang, J.-Y., 1989. Erosion and dilation of binary images by arbitrary structuring elements using interval coding. Pattern Recognition Letters 9 (3), 201–209.

Metzler, V., Thies, C., Lehmann, T., 2001. Segmentation of medical images by feature tracing in a selfdual morphological scale-space. In: Proc. SPIE-Medical Imaging. pp. 139–150.

Park, H., Yoo, J., 2001. Structuring element decomposition for efficient implementation of morphological filters. IEE Proceedings: Vision, Image and Signal Processing 148 (1), 31–35.

Robinson, K., Whelan, P., Stack, J., 2002. Segmentation of the biliary tree in mrcp data. In: Proc. SPIE Volume 4877, OPTO-Ireland: Optical Metrology, Imaging, and Machine Vision. Galway, pp. 192–200.

Sai, J., Ariyama, J., 2000. MRCP: Early Diagnosis of Pancreatobiliary Diseases. Springer-Verlag.

Salembier, P., Brigger, P., Casas, J., Pardas, M., 1996. Morphological operators for image and video compression. IEEE Transactions on Image Processing 5 (6), 881–898.

Salembier, P., Serra, J., 1995. Flat zones filtering, connected operators, and filters by reconstruction. IEEE Transactions on Image Processing 4 (8), 1153–1160.

Serra, J., 1982. Image Analysis and Mathematical Morphology. Academic Press.

Sivakumar, K., Patel, M., Kehtarnavaz, N., Balagurunathan, Y., Dougherty, E., 2000. A constant-time algorithm for erosions/dilations with applications to morphological texture feature computation. Real Time Imaging 6, 223–239.

Soille, P., 1999. Morphological Image Analysis: Principles and Applications. Springer-Verlag.

van Droogenbroeck, M., Talbot, H., 1996. Fast computation of morphological operations with arbitrary structuring elements. Pattern Recognition Letters 17, 1451–1460.

van Vliet, L., Verwer, J., 1988. A contour processing method for fast binary neighbourhood operations. Pattern Recognition Letters 7 (1), 27–36.

Vincent, L., 1993. Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms. IEEE Transactions on Image Processing 2 (2), 176–201.

Vincent, L., Soille, P., 1991. Watersheds in digital space: An efficient algorithm based on immersion simulations. IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (6), 583–598.

Woodcock, J., Loomes, M., 1988. Software Engineering Mathematics. Pitman.