On the leakage problem with the Discrete Pulse Transform decomposition

Inger Fabris-Rotelli and Gene Stoltz
Department of Statistics, University of Pretoria, 0002, Pretoria, South Africa
E-Mail: inger.fabris-rotelli@up.ac.za

***Abstract*—Connected operators act directly on connected components in an image, and though they present a strong framework for extraction of meaningful structures in an image, always suffer from the issue of leakage. The concept of leakage within any connectivity framework refers to situations in which two connected components are connected to each other via a thin, possibly long, pixel-sized connected component and are subsequently considered a single connected component. The LULU operators L_n and U_n used to derive the Discrete Pulse Transform are also connected operators and suffer from leakage. We present the Pulse Reformation algorithm to combat leakage in the pulses extracted by the DPT, making use of erosion and subsequent restricted dilations. This enables extraction of meaningful objects consisting of partial pulses of the DPT related over various scales. The examples presented illustrate a useful technique.**

I. Introduction

The concept of an axiomatic connectivity was introduced by Serra [1] and Matheron [2], for use in Mathematical Morphology. The need arose due to elements of the discrete grid on $\mathbb{Z}^2$ not satisfying a total order such as that achieved by a sequence on $\mathbb{Z}$. On $\mathbb{Z}$ one can see that we have an obvious ordering of the elements, namely x_{i+1} follows x_i and x_{i-1} precedes x_i. It is then natural to consider the elements x_{i+1} and x_{i-1} as the neighbours of x_i. However, consider the case of images defined on a discrete grid in $\mathbb{Z}^2$. Although it is natural to consider the 8 surrounding pixels for a pixel x as the neighbours, there is no immediate ordering of the neighbours as is the case in one dimension. This is because $\mathbb{Z}^2$ is only partially ordered. We can apply a raster scan to the grid, that is, starting with the first row move left to right from pixel to pixel and then repeat at next row and subsequent rows. This would however mean we have reduced the grid in $\mathbb{Z}^2$ to a sequence on $\mathbb{Z}$ and we won't have achieved a logical extension from one to two and higher dimensions. Serra and Matheron recognised this need for the concept of an axiomatic connectivity defined in Definition 1.

Definition 1: $\mathcal{C}$ is a **connectivity class** or a **connection** on $\mathcal{P}(E)$ if the following axioms hold:

(i) $\emptyset \in \mathcal{C}$
(ii) $\{x\} \in \mathcal{C}$ for each $x \in E$
(iii) For each family $\{C_i\}$ in $\mathcal{C}$ such that $\bigcap C_i \neq \emptyset$, we have $\bigcup C_i \in \mathcal{C}$.

A set $C \in \mathcal{C}$ is called **connected**.

The well-known 4- and 8-connectivity are examples of a connectivity forming a connectivity class. The concept presented in Definition 1 has been used extensively in mathematical morphology with regard to image processing applications. However, the problem of leakage has been discussed extensively in the same setting.

The concept of leakage within any connectivity framework refers to situations in which two connected components are connected to each other via a thin, possibly long, pixel-sized connected component and subsequently considered a single connected component. See Figure 1 for an illustration of this. Most commonly, this occurs due to noise or the intensity difference between objects and backgrounds. More realistically, such a connected component should be separated into the two larger connected components as these most likely represent two separate objects of the scene and have only been joined together due to noise, low resolution and such quantization effects. It results in, for example, oversegmentation or fragmentation [3], [4].

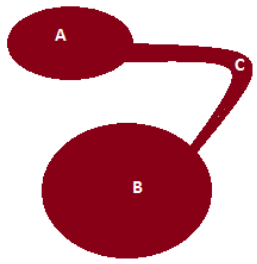


Fig. 1: An illustration of leakage in a connected component. Objects A and B are likely separate objects in reality but are connected by a thin connected component C resulting in a single connected component

Various methods have been employed to counter the problem of leakage. O'Callaghan and Bull [5] explain that leakage occurs in image segmentation due to the existence weak points in the gradient of object boundaries so that the object 'leaks' into the background. They present an improved watershed algorithm for segmentation using decimated wavelets to take care of such cases. Li and Wilson [6] make use of a multiresolution technique via the Fourier transform and a Markov random field to deal with leakage. Leakage also occurs frequently in active-contour techniques. Law and Chung [7] adapt the active-contour algorithm using a minimal weighted local variance condition to estimate where edges should be instead of allowing leakage. Lu and Bao [8] also adapt the active-

contour algorithm by requiring contours to be significantly concave or convex. Graham et al [9] also encounter leakage when developing an algorithm for human-airway segmentation, and adapt the parameters of the segmentation to be more conservative when leakage is observed. Terol-Villalobos et al [10] introduce a stopping criterion to combat leakage and obtain a result between a morphological opening and an opening by reconstruction. Wilkinson [11] defines a second generation connectivity to combat the leakage problem which occurs for all connected filters, that is, filters that operate on the connected components defined by the connectivity involved. Salembier and Oliveras [12] relax the definition of a connection (Definition 1) to define pseudo-connectivity and enable a solution to the leakage problem. Tzafestas and Maragos [13] work with multiscale connectivity obtained via their generalized connectivity measure which essentially measures the degree to which a connected component exhibiting leakage should be connected. Santillán and Herrera-Navarro [14] introduce connected viscous filters to combat leakage. Ouzounis [15] incorporates shape orientation to deal with leakage.

In this article we propose a new algorithm to combat leakage which makes use of the structure of objects in an image obtained via the Discrete Pulse Transform (DPT) [16]. The Discrete Pulse Transform based on the LULU operators for sequences was derived in [17]. Using the extension of the LULU operators L_n and U_n to functions on $\mathbb{Z}^d$ presented in [16] we present the DPT for functions in $\mathcal{A}(\mathbb{Z}^2)$. Similar to the case of sequences we obtain a decomposition of a function $f \in \mathcal{A}(\mathbb{Z}^2)$, with finite support. As usual $\mathrm{supp}(f) = \{p \in \mathbb{Z}^2 : f(p) \neq 0\}$. Let $N = \mathrm{card}(\mathrm{supp}(f))$. We derive the DPT of $f \in \mathcal{A}(\mathbb{Z}^2)$ by applying iteratively the operators L_n, U_n with n increasing from 1 to N as follows

$$DPT(f) = (D_1(f), D_2(f), ..., D_N(f)), \tag{1}$$

where the components of 1 are obtained through

$$D_1(f) = (id - P_1)(f) \tag{2}$$
$$D_n(f) = (id - P_n) \circ Q_{n-1}(f), \ n = 2, ..., N, \tag{3}$$

and $P_n = L_n \circ U_n$ or $P_n = U_n \circ L_n$ and $Q_n = P_n \circ ... \circ P_1$, $n \in \mathbb{N}$. This decomposition has the property that each component D_n in (1) is a sum of discrete pulses with pairwise disjoint supports of size n, where in this setting a discrete pulse is defined as follows.

Definition 2: A function $\phi \in \mathcal{A}(\mathbb{Z}^2)$ is called a pulse if there exists a connected set V and a nonzero real number α such that

$$\phi(x) = \begin{cases} \alpha & \text{if} \quad x \in V \\ 0 & \text{if} \quad x \in \mathbb{Z}^2 \setminus V. \end{cases}$$

The set V is the support of the pulse ϕ, that is $\mathrm{supp}(\phi) = V$. The concept of a pulse as defined in Definition 2 is similar to the idea of a flat zone from mathematical morphology. It should be remarked that the support of a pulse may generally have any shape, the only restriction being that it is connected. It follows from (2)-(3) that

$$f = \sum_{n=1}^{N} D_n(f) = \sum_{n=1}^{N} \sum_{s=1}^{\gamma(n)} \psi_{ns}, \tag{4}$$

where $\psi_{ns}, n = 1, 2, ..., \gamma(n)$ are the pulses extracted by the DPT at scale n and $\gamma(n)$ is the number of pulses of size n extracted at scale n.

The representation (4) provides a multiscale decomposition of the image f. This extracts information from the image at all possible scales and provides connected components (the pulses) which are related through through scale. We thus have multiscale objects at hand for more robust image analysis.

In Section II we present the implementation of the DPT and the proposed Pulse Reformation algorithm to deal with leakage. In Section III we provide illustrations of the technique with comparisons.

II. Algorithm

An algorithm within the DPT scale-space was developed to introduce pulse reliability and pulse 'meaning'. The algorithm defines objects within the DPT scale-space by clustering and reforming various sets of pulses. We first discuss the DPT implementation.

A. DPT Implementation

The algorithm is based on the technique developed by Laurie utilizing graph-theory [18]. The algorithm developed here makes use of two separate graphs, the Work-Graph and the Pulse-Graph. The Work-Graph is an undirected graph denoted $G_{work} = (V_{work}, E)$ representing the finite data sequence $\mathbf{x} = \{x_0, x_1, x_2, ...x_N\}$. This data sequence presents the pixel intensities in a one-dimensional array, namely $x_i = f(m_i, n_i)$ where m is the column position, n the row position and f the discrete pixel intensity function. The Pulse-Graph is a directed graph representing the extracted pulses ϕ_{ns}, where n is the scale and s the pulse number, and is denoted by $G_{Pulse} = (V_{Pulse}, A)$.

The Work-Graph is used directly in executing the DPT. The edges E represent the connectivity used in the execution of the DPT, for example 4- or 8-connectivity. The Pulse-Graph is the output of the DPT with directed edges known as arcs. The arcs show the relationship between pulses at different scales. From the data a Work-Graph is first created and then transformed into a Pulse-Graph by executing the DPT. A visual representation of the algorithm is provided in Figure 2 with a simplistic example.

In the example, the algorithm starts by using the input signal to create the Work-Graph and the basis of the Pulse-Graph. The Work-Graph is created by using each data point in the input signal as a node and two zero nodes of infinite size are added at the beginning and end of the input signal. The edges of the Work-Graph for each node are created by utilizing the required connectivity scheme to connecting the appropriate

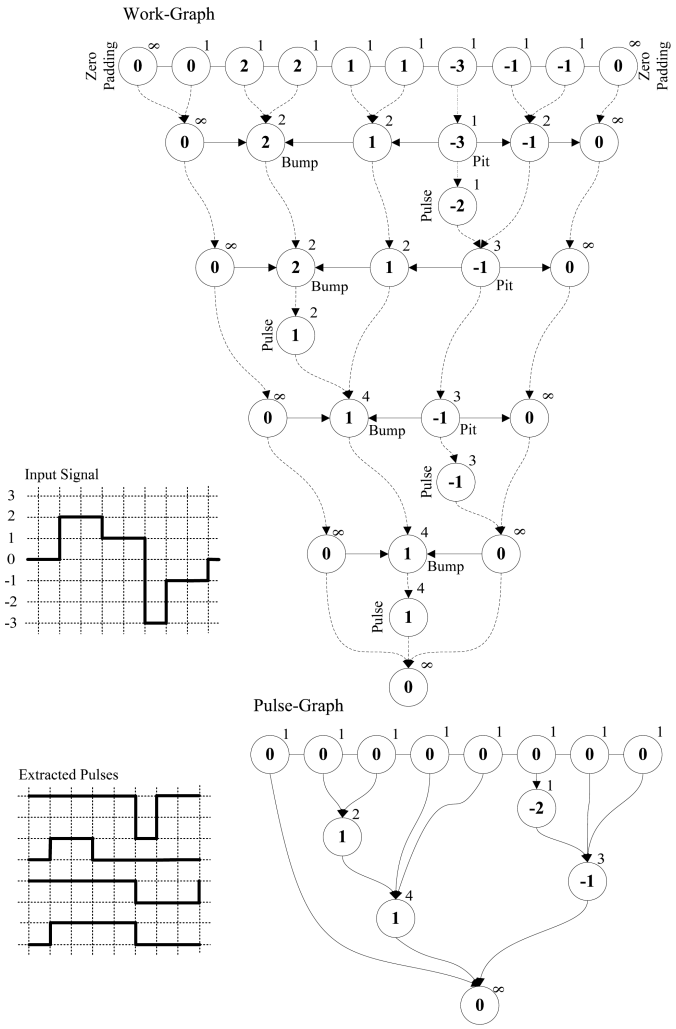


Fig. 2: Illustration of the DPT decomposition

nodes. Here a 1-dimensional wave signal is used with a 2-connectivity scheme, this scheme entails connecting each data point with its two closest neighbours.

To successfully extract the pulses their height and position must be stored and to reduce the memory requirement for the storage of the pulses, they are stored in a graph like format where the basis of this graph stores the positional information. Each node consists of arcs and a strength value. The arcs propagate the different positional information through the graph. For example, a pulse connected to the basis of the Pulse-Graph can be translated as all the base nodes connected to the pulse have a value equal to the strength of the pulse, remembering that the base nodes represents the pixel positions in an image.

The Pulse-Graph is constructed by firstly creating the basis, that is, each pixel in the image creates one node in the basis of the Pulse-Graph with a strength of zero and a size of 1. The size is one because each node only contains 1 element in the data-sequence where the strength must be zero as the sum of all nodes must be equal to the original image.

The Work-Graph is optimized by joining all connected nodes with the same value into one node retaining all relevant edges thus having one node per flat zone, and is then searched for features of every size creating a Feature-Table. A feature is defined as a local maximum(bump) or local minimum(pit) node [16]. The local neighborhood of a node is only one edge deep. The Feature-Table contains all possible features in the Work-Graph.

The decomposition is executed by searching for all size n features in the Feature-Table, $n = 1, 2, ..., N$, depending on the decomposition type. For U_nL_n or L_nU_n the pits or bumps will first be extracted respectively. Each identified feature must be reaffirmed. A feature is reaffirmed by re-checking the node, making sure it still a pit or a bump. It is possible that a feature in the Feature-Table can become a non-feature when another feature in the Feature-Table is extracted. The identified feature is extracted and a new node is created in the Pulse-Graph with the arcs connecting to the pulses that constructed the extracted pulse. By extracting a feature the relating node in the Work-Graph is merged with the node nearest in height, this node is then reaffirmed as a feature. If it is a feature the current entry in the Feature-Table is updated by changing the scale of the feature, otherwise it is removed from the Feature-table. This process is repeated by increasing the scale N each time until no more features are left in the Feature-Table, which is equivalent the final single pulse obtained by the DPT. The algorithm is implemented in the c programming environment and runs in $O(n)$ complexity.

B. Pulse Reformation

1) The Problem: Consider the four separate images in Figure 3 and regard each structure in each image as an object. One image can be created by including all four objects in it as shown in Figure 4a. The challenge is to separate these four objects from the image.

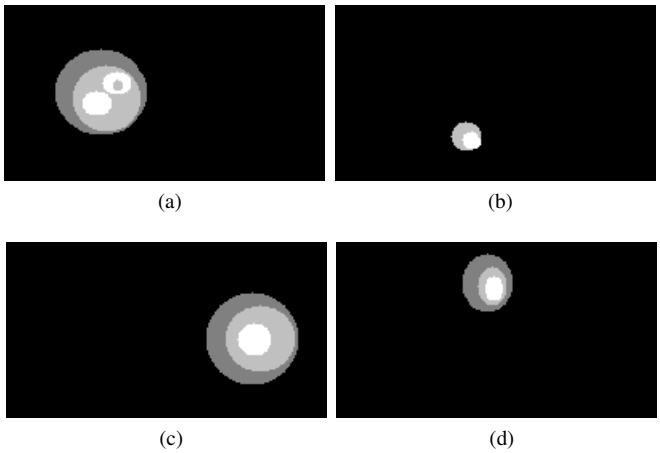


Fig. 3: The four objects which will be combined into one image and then extracted as four separate objects from the original in Figure 4a with the proposed algorithm.

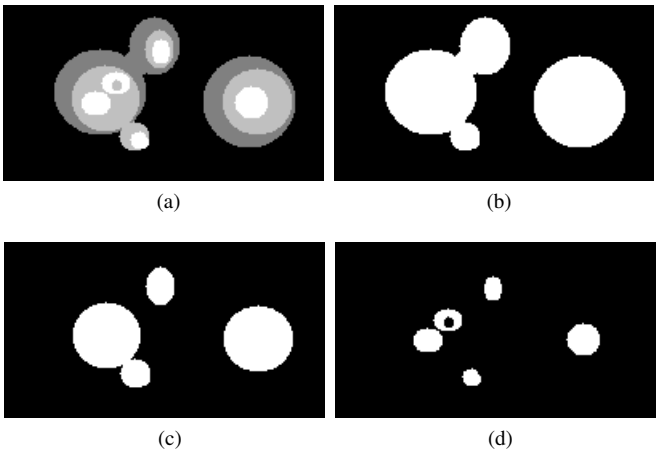


Fig. 4: Attempted extraction of the four objects given in Figure 3 from Figure 4a by using different threshold values. Thresholded images shown in (b), (c) and (d).

To extract the four objects a very simplistic method such as thresholding can be applied. This is achieved by choosing a range of intensity values where all pixel intensities outside the range becomes 0 and inside becomes 1. The number of detected objects is then directly related to the number of connected sets in the image. It can be observed in Figures 4b, 4c and 4d that all possible threshold values have been applied and none have resulted in the correct extracted connected sets.

Applying the DPT to the image a range pulse sizes chosen also provides a type of thresholding. Four ranges has been chosen and can be seen in Figure 5. Here it is also evident that the extracted connected components do not correctly present the true objects in the image.

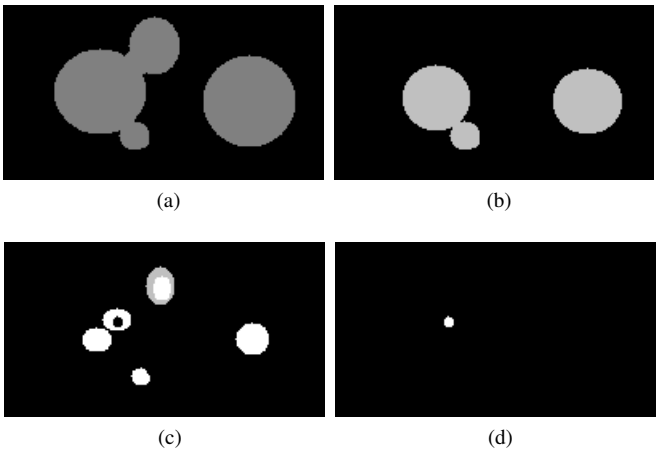


Fig. 5: Attempted extraction of the four objects in Figure 3 from Figure 4a by using different DPT pulse size ranges

By the above examples it is seen that neither thresholding in the intensity domain or in the DPT scale-space has the desired effect of successfully extracting the objects. We now present an algorithm which improves the reliability of the pulses in the DPT by removing the effect of leakage so that the pulses more accurately represent objects in the image.

2) The Solution: Within the DPT scale-space it is easy to see that an assumption of each pulse of an object containing the medial axis [19] of the pulse smaller but closest in scale of the same object is justified due to the theoretical results in [16]. The medial axis is equivalent to the morphological skeleton. This assumption is made to provide a framework for excluding texture from a shape. Texture is mostly the collection of small pulses on a much larger pulse so that it is evident that only the texture on the medial axis will be preserved and all other small pulses will be regarded as noise. We would like to separate objects that are incorrectly joined by leakage. This is achieved by dividing the medial axis of the joined objects into separate medial axis each containing one object. It is assumed that most medial axis has only one center point. To approximate the center point of a medial axis from an object, the object can be eroded consecutively until one element remains. As objects differ in size a set of joined objects can not be eroded until only one element remain as the other objects will then be lost. A set of joined objects are thus eroded until a maximum number of connected sets have been created. These connected sets, called eroded sets of the pulse, are assumed to be the approximations to the medial axis of each object in the set of joined objects. Algorithm 1 (provided in the appendix) shows the process of eroding a pulse such that the number of connected sets remaining is a maximum. It creates a binary image from the pulse. This image is then eroded with the smallest compact structuring element, a 3×3 sized element. After each erosion the number of connected components is checked and the maximum number of connected components is saved for later usage.

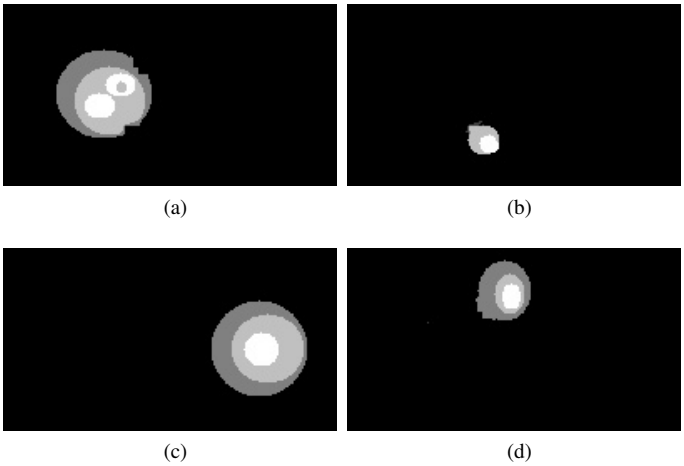
Although all the medial axis centers have been approximately located the remaining elements in the pulse must be assigned to one of these sets, each containing one medial axis. A method to reconstruct a binary image from a medial axis is to utilize morphological openings [20], which is an erosion followed by a dilation. An equivalent approach is followed here where each set is dilated within the pulse boundaries until all elements in the pulse has been assigned to a region. Algorithm 2 (provided in the appendix) shows how each eroded set in each pulse gets dilated on a ratiometric merit until all elements in the pulse have been assigned to an eroded set (region).

Another problem arises when two sets are dilated and the resulting dilations have a non-zero intersection. An element cannot belong to more than one set. To prevent this a ratiometric merit system is implemented. The eroded set with the highest ratio gets dilated first. This ratio is calculated by dividing the cardinality of the dilated eroded set with the cardinality of the set before it was dilated. Using the ratio is very important as this gives an approximation of how centered the medial axis is. However, it also hinders the dilation through leakage areas in the pulse resulting in the desired effect of eliminating leakage. Algorithm 3 (provided in the appendix) creates the regions in each pulse by utilizing the relative eroded

sets of the pulse. The cardinality of each eroded set is recorded before they are dilated. The dilation of a eroded set happens within the boundaries of the pulse, excluding the elements which is already assigned to other eroded sets. The ratio of the cardinality before and after the dilation is calculated. The elements of the eroded set with the highest ratio gets assigned too that specific eroded set. This process is repeated until all elements in the pulse have been assigned.

At this point all the pulses are divided into regions and using the Pulse-Graph, all the regions sharing a pulse also shares the same arcs. Each region must have its own unique set of arcs as it is assumed that each region is a unique object within the pulse. By taking one pulse and starting at a region in the pulse, all regions in pulses connected to the current pulse through arcs must be evaluated. To determine whether two regions are connected the related eroded set of the larger pulse must intersect with the smaller pulse. Algorithm 4 (provided in the appendix) traverses through each region created previously and determine the current region's connected regions. Each region has a relative pulse and each pulse has other pulses connected to it, which in turn have their own regions. The regions in the connected pulses are possible connected regions of the current region. The algorithm traverses through all these possible regions to determine whether they contain the approximate medial axis of the current region. If the region being tested contains the approximate medial axis it become a connected region of the current region.

All the newly created regions can now be seen as new pulses in the Pulse-Graph. This algorithm changes the structure of the original DPT, however more meaningful objects can be extracted from the image setting the scene for object detection and tracking.



(a) (b) (c) (d)

Fig. 6: The four objects of Figure 3 extracted from Figure 4a by the using Pulse Reformation algorithm

3) Cracking the problem: The algorithm presented in Section II-B2 was applied to the discussed problem in Section II-B1 with the results shown in Figure 6. It can be seen that there is some leakage onto other objects where the object is reconstructed from the approximated medial axis but all four objects was successfully extracted without applying any type of threshold or additional processing.

III. Examples

A. Text Removal

A typical problem in image processing is the successful removal of letters imposed on an image without influencing the structures not directly related to the letters. An image with some lettering is shown in Figure 7. In the image it is clear that there will be a leakage problem where the 'T' touches the horizon.

Fig. 7: An image with imposed text

To remove text with the DPT and Pulse Reformation two automatic assumptions are made. Each letter consists of a flat-zone and an approximate range for the number of elements in each flat zones is known. The Pulse Reformation creates objects with specific properties. For text one can expect to see an object with a large flat zone on the top followed with a few pulses of approximately the same size which is then supported by the background pulses. The result is shown in Figure 8.

Fig. 8: Using the Pulse Reformation algorithm the lettering can be removed from Figure 7

It is clear from Figure 8 that the text was successfully removed and that the 'T' has been successfully separated from the mountain. Visually the image did not loose any other detail. It is possible that other flat zones approximately the same size as the text were removed but this is unobservable. There is however a small change where the 'T' was. It is observable that a small part of the mountain has also been removed. This is due to the creation of the regions and the ratiometric merit. The medial axis of the two different object is only approximated and then grown from there thus that part has grown in favour of the 'T' and not the mountain. Evaluating the image, a large flat zone can be observed where the text was removed. This flat zone assumes an arbitrary intensity value approximately equivalent to the mean value of the text background. This also means that in a highly texture

environment the text will still be evident as it is not replaced with texture but only an approximate mean value of the neighbouring texture area. Further algorithmic developments could improve this technique. However, this example clearly shows the power of Pulse Reformation and a strong solution to the leakage problem.

B. Object Extraction

The algorithm main aim is to extract meaningful objects from an image via the DPT. To test this, the DPT and Pulse Reformation algorithm was applied to an image of blood cells shown in Figure 9a. The 8 strongest objects were extracted and are shown in Figure 9. Here, the object strength refers to the number of pulses it contains since the most salient structures in an image are those that are present over a wide range of scales [21].

Choosing an appropriate range for the objects required, such as the expected size of the cells, the objects in Figures 9b to 9g are easily extracted with the algorithm. Examining the extracted objects it is evident that the objects are not a true presentation of the original observed objects. The extracted objects are circular without the inside hole, with jagged edges and small missing groups of pixels. The inside hole is excluded from the object as it is treated as texture, the jagged edges are formed where a pulse is divided into multiple regions, and the pixels get lost when the approximated medial axis is dilated and no dilation can fill the pixel. To involve texture on objects, one can analyze the number and size of objects formed on top of the current object. If there are many objects of approximately the same size one can assume that it is texture and include it in the final object if necessary. Another addition is to calculated the center of mass of objects created on top of one another and if the centers are close together the separate objects can be combined to form one. One can also observe that the extracted objects have drastic variable intensity differences where it should be approximately equal. This is evidence of the formation of a region and occurs near other possible objects.

The two larger objects in Figures 9h and 9i are a good example where pulse leakage occurred and it was assumed to form one large object. By example Figures 9b and 9c forms part of the object shown in Figure 9h and can be seen as texture on the object. To distinguish between such cases an approximated size for an object is required.

The Pulse Reformation can be compared to similar techniques such as the extraction of λ-connected components [14] which must be used in conjunction with a thresholding technique In this case Otsu's method [22] will be utilized. The λ-connected components are those in which the center of a disk structuring element of radius λ can be moved along a continuous path throughout the connected component such that the entire disk stays within the domain of the connected component. A few examples are shown in Figure 10.

It can be observed in the samples that the λ-connected components do not successfully extract the correct objects. In Figure 10c a manually tuned threshold which provided the best results was used. With this image the correct number of objects can be extracted by using a known cardinality range of the connected components. Figure 10 demonstrates that it

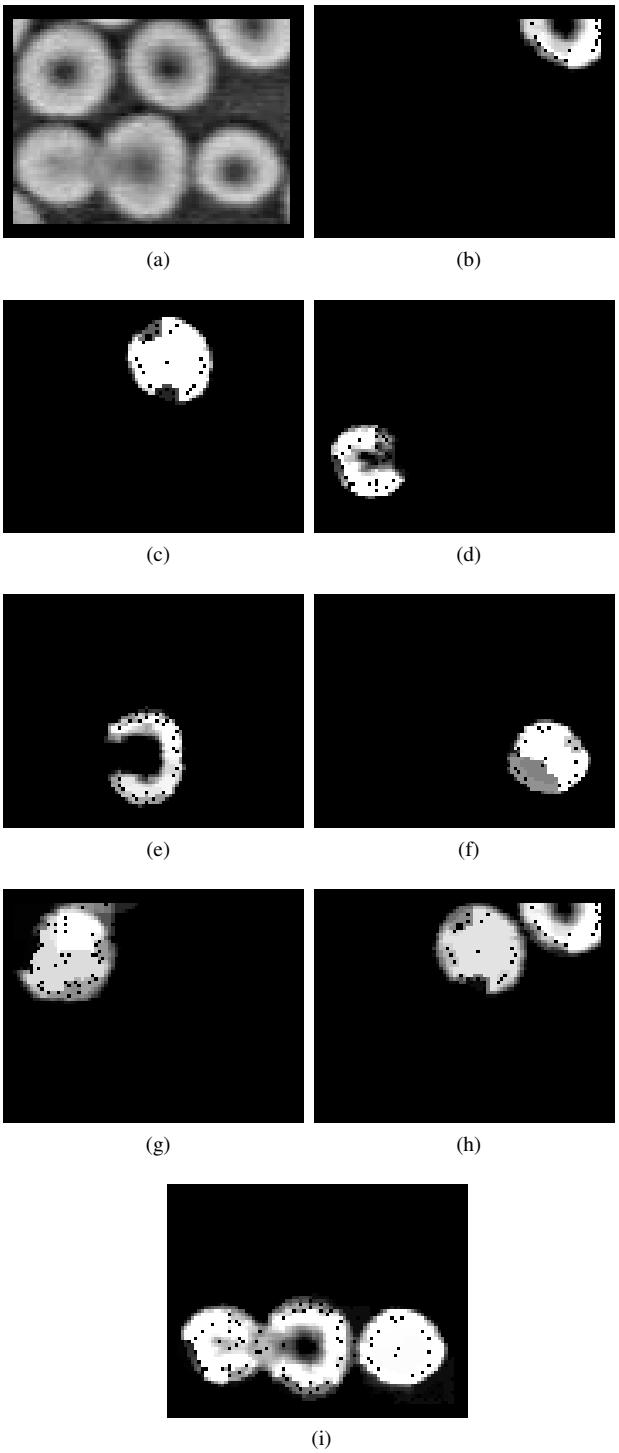
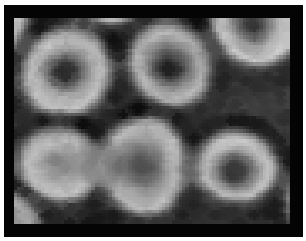
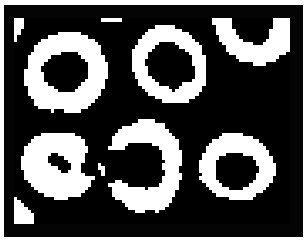


Fig. 9: The Pulse Reformation algorithm is used to extract possible objects (blood cells) within Figure 9a. The 6 blood cells are successfully extracted in addition to two other possible larger objects.

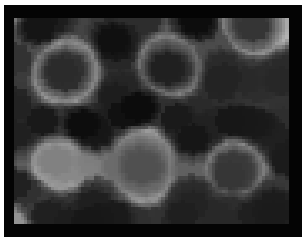


(a) $\lambda = 1$

(b) $\lambda = 1$ Thresholded

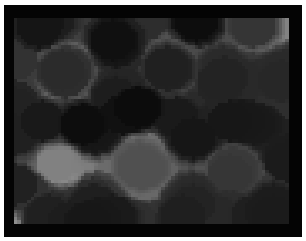


(c) $\lambda = 1$ Manual Threshold



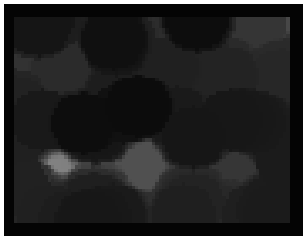
(d) $\lambda = 4$

(e) $\lambda = 4$ Thresholded



(f) $\lambda = 6$

(g) $\lambda = 6$ Thresholded



(h) $\lambda = 9$

(i) $\lambda = 9$ Thresholded

Fig. 10: λ-connected components are created and thresholded in an attempt to extract the connected sets which presents the objects(blood cells) in Figure 9a. Only when using a manual threshold was this achieved.

is clear that the Pulse Reformation is successful at extracting objects without the use of any thresholding regarding pixel intensity. The fact that thresholding is not required gives the algorithm a large advantage, the only information still required is an approximate range of the size of the objects in question.

IV. Conclusion

Connected operators [23] act directly on connected components, and though they present a strong framework for extraction of meaningful structures in an image, always suffer from the issue of leakage defined in Section I. The LULU operators L_n and U_n used to derive the Discrete Pulse Transform are also connected operators and suffer from leakage. We have presented the Pulse Reformation algorithm to combat leakage in the pulses extracted by the DPT. This enables extraction of meaningful objects consisting of pulses of the DPT related over various scales. The examples presented illustrate a useful technique which will be theoretically investigated as well as refined in future research.

V. Appendix

Algorithm 1 Approximate the medial centers

```
for (each pulse[i]) {
  BinaryI = Create binary image of pulse;
  do {
    BinaryI = eroded BinaryI;
    eroded_k = amount of connected sets in
        BinaryI;
  } while (eroded_k is not maximum);
  for (each eroded_k) {
    Eroded set[i][k] = Connected set[k] in
        BinaryI;
  }
}
```

Algorithm 2 Create regions in a pulse

```
for (each pulse[i]) {
  for (each Eroded Set[i][k]) {
    RegionI[k] = Eroded Set[i][k];
  }
  TotalRegions = Union of all RegionI[k];
  TotalRegionSize = Cardinality of
      TotalRegions;
  PulseSize = Cardinality of pulse[i];
  while (TotalRegionSize != PulseSize)
    Dilate the RegionI[k] with largest
                  Ratio[k];
    TotalRegions = Union of all RegionI[k];
    TotalRegionSize = Cardinality of
        TotalRegions;
  }
}
```

Algorithm 3 Dilate RegionI[k] with largest Ratio

```
for (each Eroded Set[i][k]) {
  MaskI = Union of all RegionI excluding
      RegionI[k];
  MaskIPulse = Pulse[i] excluding MaskI;
  DilatedI[k] = Dilation of RegionI[k]
      intersecting with MaskIPulse;
  Ratio[k] = Cardinality(DilatedI[k]) /
      Cardinality(RegionI[k]);
}
Max_k = k value related to maximum value of
    Ratio[k];
RegionI[Max_k] = Union of RegionI[Max_k] and
    DilatedI[Max_k];
```

Algorithm 4 Connecting regions through arcs.

```
for (each pulse[i]){
  for (each RegionI[i][k]){
    for (each pulse connected by Arc[i][m]){
      for (each RegionI[m][n]){
        If (Eroded Set[i][k] Intersects with
            RegionI[m][n]){
          Add Arc from RegionI[m][n] to
                       Region[i][k];
        }
      }
    }
  }
}
```

REFERENCES

[1] J. Serra, *Image Analysis and Mathematical Morphology, Volume II: Theoretical Advances*. London: Academic Press, 1988, ch. Mathematical Morphology for Boolean Lattices.

[2] G. Matheron, *Image Analysis and Mathematical Morphology, Volume II: Theoretical Advances*. London: Academic Press, 1988, ch. Filters and lattices.

[3] G. Ouzounis and M. Wilkinson, "Countering oversegmentation in partitioning-based connectivities," in *Proc. Int. Conf. Image Processing*, 2005, pp. 844–847.

[4] M. Wilkinson, "Attribute-space connectivity and connected filters," *Image Vis. Comput.*, vol. 25, pp. 426–435, 2007.

[5] R. O'Callaghan and D. Bull, "Combined morphological-spectral unsupervised image segmentation," *IEEE Transactions on Image Processing*, vol. 14, no. 1, pp. 49–62, 2005.

[6] C.-T. Li and R. Wilson, "Image segmentation based on a multiresolution bayesian framework," in *Proceedings of the 1998 International Conference on Image Processing, ICIP*, vol. 3, 4-7 October 1998, pp. 761–765.

[7] W. Law and A. Chung, "Minimal weighted local variance as edge detector for active contour models," in *Computer Vision-ACCV 2006, Lecture Notes in Computer Science*, vol. 3851/2006. Springer-Verlag, Berlin, Heidelberg, 2006, pp. 622–632.

[8] H. Lu and S. Bao, "Physical modeling techniques in active contours for image segmentation," submiteed 22 June, last revision 30 June 2009 2009, cornell University Library Archives.

[9] M. Graham, J. Gibbs, and W. Higgins, "Robust system for human airway-tree segmentation," in *Medical Imaging 2008: Image Processing, Proceedings of SPIE*, J. Reinhardt and P. Pluim, Eds., vol. 6914 69141J-1, 2008.

[10] I. Terol-Villalobos, J. M.-S. nez, and S. Canchola-Magdaleno, "Image segmentation and filtering based on transformations with reconstruction criteria," *Journal of Visual Communication and Image Representation*, vol. 17, pp. 107–130, 2006.

[11] M. Wilkinson, "Connected filtering by reconstruction: basis and new advances," in *Proceedings of 15th IEEE International Conference on Image Processing, ICIP*, 12-15 October 2008 2008, pp. 2180–2183.

[12] P. Salembier and A. Oliveras, *Mathematical Morphology and its Applications to Images and Signal Processing*. Kluwer Academic, 1996, ch. Practical extensions of connected operators, pp. 97–110.

[13] C. Tzafestas and P. Maragos, "Shape connectivity: multiscale analysis and application to generalized granulometries," *Journal of Mathematical Imaging and Vision*, vol. 17, pp. 109–129, 2002.

[14] I. Santillán, A. Herrera-Navarro, J. M.-S. nez, and I. Terol-Villalobos, "Morphological connected filtering on viscous lattices," *Journal of Mathematical Imaging and Vision*, vol. 36, pp. 254–269, 2010.

[15] G. Ouzounis, "Generalized connected morphological operators for robust shape extraction," PhD Thesis, University of Groningen, 2009.

[16] R. Anguelov and I. N. Fabris-Rotelli, "LULU operators and discrete pulse transform for multi-dimensional arrays," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 3012–3023, 2010.

[17] C. Rohwer, *Nonlinear Smoothers and Multiresolution Analysis*. Birkhäuser, 2005.

[18] D. Laurie, "The roadmaker's algorithm for the discrete pulse transform," *IEEE Transactions on Image Processing*, vol. 20, no. 2, pp. 361–371, 2011.

[19] H. Blum, *Models for the Perception of SPeech and Visual Form*. Cambridge: MIT Press, 1967, ch. A Transformation for Extracting New Descriptors of Shape, pp. 362–380.

[20] J. Serra, *Image Analysis and Mathematical Morphology*. Orlando, FL, USA: Academic Press, Inc., 1983.

[21] T. Lindeberg and K. Mardia, "Scale-space theory: A basic tool for analyzing structures at different scales," *Journal of Applied Statistics*, vol. 21, no. 1/2, pp. 225–271, 1994.

[22] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[23] U. Braga-Neto and J. Goutsias, "Grayscale level connectivity: Theory and applications," *IEEE Transactions on Image Processing*, vol. 13, no. 12, pp. 1567–1580, 2004.