Connectivity Preserving Transformations for Higher Dimensional Binary Images

Anvesh Komuravelli

Computer Science and Engineering Department,
Indian Institute of Technology, Kharagpur, Kharagpur-721302, India.

Arnab Sinha

Dept. of Electrical Engineering, Princeton University,
Princeton, New Jersey, USA-08544.

Arijit Bishnu

Advanced Computing and Microelectronics Unit, Indian Statistical Institute,
203 B. T. Road, Kolkata-700108, India.

Abstract

An N-dimensional digital binary image (I) is a function $I : \mathbb{Z}^N \rightarrow \{0, 1\}$. I is $\mathbf{B}_{3^N-1}, \mathbf{W}_{3^N-1}$ connected if and only if its black pixels and white pixels are each $(3^N - 1)$-connected. I is only $\mathbf{B}_{3^N-1}$ connected if and only if its black pixels are $(3^N - 1)$-connected. For a 3-D binary image, the respective connectivity models are $\mathbf{B}_{26}, \mathbf{W}_{26}$ and $\mathbf{B}_{26}$. A pair of $(3^N - 1)$-neighboring opposite-valued pixels is called *interchangeable* in a N-D binary image I, if reversing their values preserves the original connectedness. We call such an interchange to be a $(3^N - 1)$-local interchange. Under the above connectivity models, we show that given two binary images of n pixels/voxels each, we can transform one to the other using a sequence of $(3^N - 1)$-local interchanges. The specific results are as follows. Any two $\mathbf{B}_{26}$-connected 3-dimensional images I and J each having n black voxels are transformable using a sequence of $O((c_1 + c_2)n^2)$ 26-local interchanges. Here, c_1 and c_2 are the total number of 8-connected components in all 2-dimensional layers of I and J respectively. We also show bounds on $\mathbf{B}_{26}$ connectivity under a different interchange model as proposed in [3]. Next, we show that any two *simply connected* images under the $\mathbf{B}_{26}$,$\mathbf{W}_{26}$ connectivity model and each having n black voxels are transformable using a sequence of $O(n^2)$ 26-local interchanges. We generalize this result to show that any two $\mathbf{B}_{3^N-1}$,$\mathbf{W}_{3^N-1}$-connected N-dimensional *simply connected* images each having n black pixels are transformable using a sequence of $O(Nn^2)$ $(3^N - 1)$-local interchanges, where $N > 1$.

Key words: binary image transformation, connectedness, local interchange, IP-equivalence

1 Introduction

An N-dimensional digital binary image (I) is a function $I : \mathbb{Z}^N \to \{0, 1\}$. We will denote any element in $\mathbb{Z}^N$ by an N-dimensional pixel $(d_1, d_2, \ldots, d_N)$. We will sometimes refer to it as a *high dimensional pixel*. Particularly, any element in $\mathbb{Z}^3$ ($\mathbb{Z}^2$) is called a *voxel* (*pixel*). Any N-dimensional pixel p is black (white) if $I(p) = 1$ ($I(p) = 0$). We consider finitely many black N-dimensional pixels from $\mathbb{Z}^N$ in I.

1.1 Connectivity and Interchange model

We call two pixels $(d_1, d_2, \ldots, d_N)$ and $(d'_1, d'_2, \ldots, d'_N)$ to be (3^N-1)-neighbors if and only if $|d_i - d'_i| \leq 1$ for all $1 \leq i \leq N$ and not every d'_i is the same as d_i. This neighborhood is defined based on the recurrence $f(N) = 3f(N-1) + 2$ for $N > 2$ with the initial condition $f(2) = 8$, where $f(N)$ denotes the number of neighbors in the N-th dimension. The above definition induces a graph G_{3^N-1} whose vertex set is $\mathbb{Z}^N$ and there exist edges between two lattice points satisfying the above condition of $(3^N - 1)$-neighborhood. In an N-D binary image I, $\mathbf{B}_{3^N-1}(I)$ is a sub-graph of G_{3^N-1} induced by the N-dimensional black pixels in I. Similarly, $\mathbf{W}_{3^N-1}(I)$ is a sub-graph of G_{3^N-1} induced by the N-dimensional white pixels in I. We say that an N-dimensional binary image I is $\mathbf{B}_{3^N-1}$ connected if the graph $\mathbf{B}_{3^N-1}(I)$ is connected. We say that an N-dimensional binary image I is $\mathbf{B}_{3^N-1}, \mathbf{W}_{3^N-1}$ connected if both the graphs $\mathbf{B}_{3^N-1}(I)$ and $\mathbf{W}_{3^N-1}(I)$ are connected. There can be other neighborhoods based on some other functions of N capturing the information of the number of dimensions (coordinates) that are allowed to differ. Note that, in the neighborhood defined above, all the N dimensions can be different (by at most 1). As an example, we call two pixels $(d_1, d_2, \ldots, d_N)$ and $(d'_1, d'_2, \ldots, d'_N)$ to be $2N$-neighbors if and only if $|d_i - d'_i| = 1$ for exactly one i such that $1 \leq i \leq N$. The recurrence in this case is $f(N) = f(N-1) + 2$ with the initial condition $f(1) = 2$.

More particularly, in a two dimensional binary image, we call two different pixels (x_1, y_1) and (x_2, y_2) to be 8-neighbors if and only if either or both of the conditions $|x_1 - x_2| \leq 1$, $|y_1 - y_2| \leq 1$ hold. Also, we call two different pixels (x_1, y_1) and (x_2, y_2) to be 4-neighbors if and only if $|x_1 - x_2| = 1$ or $|y_1$

Email addresses: anvesh@cse.iitkgp.ernet.in (Anvesh Komuravelli), sinha@princeton.edu (Arnab Sinha), arijit@isical.ac.in (Arijit Bishnu).

$- y_2| = 1$. Note that these conditions for being 8-neighbors and 4-neighbors are equivalent to $(x_1 - x_2)^2 + (y_1 - y_2)^2 \leq 2$ and $(x_1 - x_2)^2 + (y_1 - y_2)^2 \leq 1$, respectively. Similarly, in a three dimensional binary image, we call two voxels (x_1, y_1, z_1) and (x_2, y_2, z_2) to be 26-neighbors (6-neighbors) if and only if $(x_1-x_2)^2 + (y_1-y_2)^2 + (z_1-z_2)^2 \leq 3$ ($(x_1-x_2)^2 + (y_1-y_2)^2 + (z_1-z_2)^2 \leq 1$). As a special case for two dimensional images, some other connectednesses like $(\mathbf{B}_4, \mathbf{W}_4)$, $(\mathbf{B}_4, \mathbf{W}_8)$ and $(\mathbf{B}_8, \mathbf{W}_4)$ can be defined in addition to $(\mathbf{B}_8, \mathbf{W}_8)$ [1]. For a 3-D binary image, the connectedness model we consider is $\mathbf{B}_{26}, \mathbf{W}_{26}$.

A pair of $(3^N - 1)$-neighboring opposite-valued pixels $\langle p, q \rangle$ is called *interchangeable* in an N-D binary image I, if reversing their values preserves the topology of the image [6]. We will call such an interchange to be a (3^N-1)-local interchange. For a 2-D binary image I, a pair of 8-neighbor opposite-valued pixels is called *interchangeable* if reversing their values preserves the topology of the image [5,6]. The interchange does not affect the number of 0s and 1s in I. Two 2-D binary images I and J are called *IP-equivalent* or *transformable* [5,6] if there exists a sequence of binary images $I = I_0, I_1, \ldots, I_i, \ldots, I_k = J$ such that any I_i $(1 \leq i \leq k)$ can be obtained from I_{i-1} by reversing an *interchangeable* pixel pair. This definition of *IP-equivalence* as given in [5,6] can be generalized for other connectivity models in 2-D as well as for higher dimensional images.

Our work in this paper in a way generalizes some of the results of Rosenfeld and Nakamura [6] and Bose et al.[1] for two dimensional binary images to higher dimensional binary images. Below, we review these results of two dimensional connectivity preserving pixel transformation.

1.2 Prior Work

Rosenfeld and Nakamura [6] proved the conjecture made in [5] that if two binary images I and J have two simply connected sets S and T respectively of the same number of 1s, then I and J are *IP-equivalent*. In a recent comprehensive work that also deals with the combinatorial bounds on the number of interchanges, Bose et al. [1] generalized the results in [6]. They showed that for any $(a, b) \in \{(4, 8), (8, 4), (8, 8)\}$, any two $\mathbf{B}_a, \mathbf{W}_b$-connected images I and J each with n black pixels differ by a sequence of $O(n^2)$ interchanges. This is optimal within a constant factor as converting a horizontal image (a horizontal line, to be precise) to a vertical image takes $\Omega(n^2)$ interchanges. The interchanges considered are 8-local, i.e. two opposite valued pixels can be interchanged if they are 8-neighbors and reversing them does not change the topology of the image. The corresponding result for two $\mathbf{B}_4, \mathbf{W}_4$ connected images is $O(n^4)$ though here also the same horizontal to vertical conversion takes $\Omega(n^2)$. The problem of bridging the gap between $O(n^4)$ and $\Omega(n^2)$ is still

open. The interchanges considered by Bose et al. [1] and Rosenfeld and Nakamura [6] maintain connectivity of both foreground and background. The only restriction they put on I and J is that both have to be simply connected. Bose et al. [1] ensured the simply connectedness by pointing out that a 2-D binary image I is $\mathbf{B}_a, \mathbf{W}_8$-connected, $a \in \{4, 8\}$, if and only if $\mathbf{B}_a(I)$ is connected and $\mathbf{B}_4(I)$ does not contain a cycle C such that there exists a white pixel inside C in I. Similarly, for $\mathbf{B}_a, \mathbf{W}_4$-connected model, $\mathbf{B}_8(I)$ does not contain a cycle as above.

This sort of transformation problems has motivation in robotics [2] where researchers are interested in the number of moves needed in going from a configuration to another under some restrictions in the movement patterns. Under a more restricted and thus easier connectivity model but a complex and difficult interchange rule, Dumitrescu and Pach [3] show that any two $\mathbf{B}_4$ connected images are apart by $O(n^2)$ interchanges where an interchange takes place between two 8-neighbor pixels such that the image obtained after the interchange is still $\mathbf{B}_4$ connected. Though Dumitrescu and Pach consider modular metamorphic systems in terms of motion planning in [3], similarity to pixels is straightforward.

1.3 Our Work

For issues related to connectedness in digital topology [4], a *hole*, which is a set of connected component (a maximal connected subgraph) of white pixels "completely enclosed" by a connected set of black pixels, cannot have connection to any white pixel not in its connected component. The connectedness model should ensure this. The work of Rosenfeld and Nakamura [6] and Bose et al. [1] rule out the existence of *holes* because they require the images to be simply connected. Motivated by this and the model of Dumitrescu and Pach [3], we first consider a simplistic model of connectedness namely $\mathbf{B}_{26}$ whose definition, only involving black pixels/voxels, is impervious to the existence or non-existence of *holes*. We show that a 3-D binary image consisting of n voxels can be transformed to another 3-D binary image consisting of n voxels using 26-local interchanges under the $\mathbf{B}_{26}$-connectivity model. We also show that two such 3-D binary images are transformable under $\mathbf{B}_{26}$-connectivity using a different interchange model called *single backbone condition*. This *single backbone condition* has been defined in [3] for 2-D. The above two works were reported by us in an earlier version [7]. Next, to generalize the results of [1] and [6], we focus on $\mathbf{B}_{26}, \mathbf{W}_{26}$-connected model and further discuss results on $\mathbf{B}_{3^N-1}, \mathbf{W}_{3^N-1}$-connected model for an N-dimensional binary image. The interchange model used is (3^N-1)-local interchange. For the $\mathbf{B}_{26}, \mathbf{W}_{26}$-connected and $\mathbf{B}_{3^N-1}, \mathbf{W}_{3^N-1}$-connected models, we stick to the assumption as in [1,6] that the images under consideration are *simply connected*. To the best of our

knowledge, connectivity preserving *high dimensional pixel* transformation has not been considered earlier.

Section 2 discusses preliminaries. As mentioned already, we consider two types of *connectivity models* - $\mathbf{B}_{26}$ and $(\mathbf{B}_{3^N-1}, \mathbf{W}_{3^N-1})$ ($\mathbf{B}_{26}, \mathbf{W}_{26}$ for 3-D). The *interchange model* is $(3^N - 1)$-local interchange (26-local for 3-D) for the discussions in Sections 3, 5 and 6. The *interchange model* of *single backbone condition* for Section 4 is a bit stricter than the 26-local interchange and would be discussed in Section 4. In Section 6, we generalize the result of Section 5 to connectivity preserving *high dimensional pixel* transformation for the $\mathbf{B}_{3^N-1}, \mathbf{W}_{3^N-1}$-connected model. Sections 5 and 6 assume that the images under consideration are *simply connected*. Section 7 sums up the findings of our work. Below is a section-wise listing of the connectivity and interchange models used.

Section	Connectivity model	Interchange model
3	$\mathbf{B}_{26}$	26-local
4	$\mathbf{B}_{26}$	26-local+*single backbone condition*
5	$\mathbf{B}_{26}, \mathbf{W}_{26}$	26-local
6	$\mathbf{B}_{3^N-1}, \mathbf{W}_{3^N-1}$	$(3^N - 1)$-local

2 Preliminaries

Under the above connectivity and interchange models, we say that two binary images I and J of the same number of black pixels are called *transformable* if there exists a sequence of binary images $I = I_0, I_1, \ldots, I_i, \ldots, I_k = J$ such that any I_i $(1 \leq i \leq k)$ can be obtained from I_{i-1} by reversing an *interchangeable* pixel pair. We do this by transforming I to a linear chain of black pixels. So, it follows that J can also be transformed to a linear chain of black pixels; and the transformation of I to J can be obtained by transforming I to a linear chain of black pixels and then retracing the transformation (of J) from the linear chain of black pixels back to J. *Complexity analysis*, wherever used, denotes the number of *interchanges* between black and white pixels required for the particular algorithm.

2.1 Definitions and Notations

Let $G = (V_G, E_G)$ be a graph where V_G is the set of vertices and E_G is the set of edges. The subgraph of G induced by a set of vertices $S \subseteq V_G$ is denoted

as $G[S]$ and $G[S] = (S, E(S))$ where $E(S) = \{(u, v) \in E_G | u, v \in S\}$. A non-empty graph G is *connected* if there is a path between any pair of vertices in G. A *component* is a maximal connected subgraph of G. A *cut vertex* is a vertex of V_G whose removal disconnects G, otherwise, the vertex is *non-cut*. For a graph G and a vertex $u \in V_G$, let $A_G(u)$ denote the set of all vertices in $V_G \setminus \{u\}$ such that there exists an edge $(u, v) \in E_G$ where $v \in V_G \setminus \{u\}$. Also, $A_G[u] = A_G(u) \cup \{u\}$. The following observations [1] will also be useful. The first one is a simple observation from graph theory and the second one gives a sufficient condition for an interchange to preserve connectivity.

Observation 1 *For a graph G, a vertex $v \in V_G$ and a set of vertices $S \subseteq V_G \setminus \{v\}$, if $G[A_G(v) \cup S]$ is connected then v is not a cut vertex of G.*

Observation 2 *For a $\mathbf{B}_{3^N-1}, \mathbf{W}_{3^N-1}$-connected image I, let p be a black pixel that is not a cut vertex in $\mathbf{B}_{3^N-1}(I)$ and q a white pixel that is not a cut vertex in $\mathbf{W}_{3^N-1}(I)$. If p has a white $(3^N - 1)$-neighbor in I other than q and q has a black $(3^N - 1)$-neighbor in I other than p, then the interchange of p and q preserves the original $\mathbf{B}_{3^N-1}, \mathbf{W}_{3^N-1}$-connectivity.*

2.2 Definitions, Notations and Solution Strategy specific to $\mathbf{B}_{26}$

See Fig. 1 for this discussion. When a voxel moves because of the interchanges such that its z-coordinate remains unaffected, we use the term *pixel* also. In the body of the text, we interchangeably use the term *voxel* and *pixel*.

Layer: A 3-D image spans over some layers, where each layer contains a 2D structure.

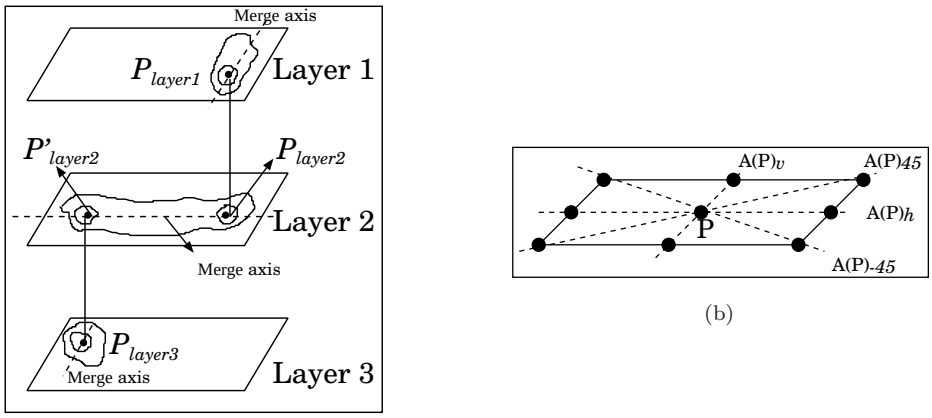
Connectivity-sensitive pixel: Consider the topmost layer (let it be *Layer 1*) of Fig. 1(a). As the image (the black component) is connected, there must exist at least one black pixel P_{layer1} which has a 26-neighbor in the layer just below it. We denote such pixels as *connectivity-sensitive* pixels. For the preservation of connectivity, one of these connectivity-sensitive pixels is not interchanged during the first part of the transformation, as the layers present below are hanging from that particular pixel of the top-layer.

Merge Axis: Merge axis ($\mathcal{M}(P)$) is a coordinate axis lying on a layer and passing through a *connectivity-sensitive pixel*. The pixels lying on the merge axis are defined to be *non-interchangeable* throughout the first part of the transformation. Figure 1(a) shows $\mathcal{M}(P_{layer1})$, $\mathcal{M}(P_{layer2})$ and $\mathcal{M}(P_{layer3})$ in *Layer 1*, *Layer 2* and *Layer 3* respectively. All the black pixels of the given 2D component are finally brought onto or *merged* on this axis using connectivity preserving interchanges. Where P is obvious, we use just $\mathcal{M}$.

Level: In a given layer and in a given connected component in that layer, a *level* is the shortest distance of a pixel of that component, from the *Merge*

Axis of the corresponding connected component.

Coordinate Axes: For any black pixel on a 2D layer, its four coordinate axes determine the direction in which the adjacent black pixels are located. The coordinate axes through pixel P in Fig. 1(b) are the following (i) $A(P)_v$ (vertical axis), (ii) $A(P)_h$ (horizontal axis), (iii) $A(P)_{45}$ (making 45° with $A(P)_h$) and (iv) $A(P)_{-45}$ (making -45° with $A(P)_h$).



(a) (b)

Fig. 1. (a) The 3D image in different layers. The adjacency between P_{layer1} and P_{layer2} (P'_{layer2} and P_{layer3}) maintains the connectivity across *Layer 1* and *Layer 2* (*Layer 2* and *Layer 3*). (b) The coordinate axes through a black pixel P.

Merge Path: Extending the concept of *Merge Axis*, a Merge Path is a path (not necessarily a straight line) on which we finally merge all the pixels.

In this problem, our fundamental strategy is to attack the 2D layers of a $\mathbf{B}_8$-connected finite binary image found in a $\mathbf{B}_{26}$-connected 3D object. In a given 2D layer, a black pixel can have at most 8 neighbors. Hence, we borrow from Bose et al. [1] the idea of transforming any 2D binary image into a *vertical* image, ensuring that the image preserves connectivity during the transformation. However, we cannot directly adopt the strategy in [1] since the vertical image produced in the 2D plane is *unique* and in our case might snap the connectivity between two layers. A general case of the problem may have the images I and J such that, each layer has more than one connected component of black pixels.

2.3 Definitions, Notations and Solution Strategy specific to $\mathbf{B}_{3^N-1}, \mathbf{W}_{3^N-1}$

The main idea of Bose et al. [1] is to reduce any 2D image to a vertical image which is a one dimensional structure. Any two binary images with n black pixels each are now equivalent via this vertical image of n black pixels. This they do with the help of a potential function which reduces at each step of interchange by at least one. Thus, the potential function helps in reducing the

dimension of the black pixels in the binary image from two to one. We carry this idea forward. Given an N-D binary image, we can successively reduce its dimension till we reach the base case of a two dimensional image from where the method of Bose et al. [1] works.

Consider any *high dimensional pixel* $(d_1, d_2, \ldots, d_N)$ in the image I. Without loss of generality, assume that the minimum value of d_1 for any black pixel is 0 and that of all black pixels with $d_1 = 0$, the minimum value of d_2 is 0. To every black pixel $(d_1, d_2, \ldots, d_N)$ we assign a *potential function*, $\phi(p) = d_1 + 2(n - d_2)$ and define the potential of the image as $\phi(I)$ which is the sum of the potentials of all the black pixels in I. We show that using $(3^N - 1)$-local interchanges we can bring down $\phi(I)$ by at least one in each iteration. Let I_i be the image formed after i iterations. As the maximum possible absolute values of d_1 and d_2 are n each (there are only n black pixels in I), we have $\phi(p) \leq 5n$ for any black pixel, p. So, $\phi(I) \leq 5n^2$. Thus, in i $(= O(n^2))$ iterations, $\phi(I_i)$ crosses zero towards the negative side. Now, $\phi(I_i)$ can be negative only if $d_1 < 0$ for some black pixel. Given that $d_1 \geq 0$ for every black pixel in I, this can only happen if there is an interchange involving a black pixel with $d_1 = 0$ in I_{i-1}. But the algorithm we present ensures that this does not happen and it exits when every black pixel has $d_1 = 0$, giving an $(N-1)$-dimensional image. We repeat the algorithm until we get a 2-dimensional image from which the method in [1] works. This is the main idea behind our generalizations.

As an example, consider the following for a 3-D binary image. As we are going from 3D to 2D, we take two (x and z) of the three coordinates (x, y and z). We make the assumption that the minimum x-coordinate of any black voxel is 0 and out of all such black voxels, the minimum z-coordinate is 0. Now, define the potential of a voxel $v \in I$ as $\phi(v) = x + 2(n - z)$ and from that define $\phi(I)$. We would show that using 26-local interchanges, we can at each step of our iteration bring down $\phi(I)$ by at least one. We stop when we get $x = 0$ for all black voxels. Thus, we get a 2-D binary image from which the method in [1] works.

3 The Strategy for Voxel Transformation in $\mathbf{B}_{26}$ model

Define a graph $G = (V_G, E_G)$, such that (i) each connected component in any layer corresponds to a node in V_G, and (ii) for any two connected components, C_1 and C_2, if there is at least one pair of voxels (u, v) which are $\mathbf{B}_{26}$ adjacent, with $u \in C_1$ and $v \in C_2$, then we have an edge. It is easy to see that, as the black pixels in the original image I are connected, G is connected. Let $G' = (V_G, E'_G)$ be any spanning tree of G. We know from the definition of a spanning tree that, as long as G remains connected G' also remains connected, thus satisfying the principal constraint behind the transformation. So, it is

sufficient to consider G', instead of G. Also, we know that every spanning tree has *at least one* node whose degree is equal to one. Before we discuss the actual algorithm we discuss below a construction which is frequently used in the algorithm.

3.1 Construction of 2D Linear Chains

Given a node in G' with degree one, we need to consider only one *connectivity-sensitive pixel* in the component represented by the node to preserve the connectivity. So, a *Merge Axis* can be any *coordinate axis* passing through that pixel. Now, the rest of the black pixels (which do not originally lie on the merge axis) are interchanged preserving the connectivity such that they finally appear as a linearly connected chain along the merge-axis.

The strategy can be outlined as follows. We compress the 2D region, step by step, from the boundary, simultaneously expanding on the merge axis, $\mathcal{M}$. Here, boundary refers to the outer boundary of the B_8-connected 2D region under consideration. Ultimately, we have the linear connected chain on $\mathcal{M}$ of all the pixels originally in the 2D plane. Now, we describe our algorithm.

Take a *non-cut* pixel (if any) on the boundary, other than those on $\mathcal{M}$. Clearly, its removal doesn't disconnect the rest of the black region. Hence, we move it along the boundary until we first reach $\mathcal{M}$, interchanging with the white pixels that come in the way. This clearly maintains connectivity of the black pixels. Place it on $\mathcal{M}$, by interchanging with the white pixel already present.

We repeat the above process till all the *non-cut* pixels are exhausted. Now, we are left with only *cut* pixels on the boundary (if any).

Figure 2(a) shows the part of the original image, which is of concern (one layer). The movement of the *non-cut* pixel P along the boundary to the merge axis $\mathcal{M}$ is shown in Fig. 2(b) to Fig. 2(d).

Lemma 1 *Consider the situation when all the black pixels on the boundary, not on $\mathcal{M}$, are* cut *pixels. Also consider a part of the boundary which starts and ends on $\mathcal{M}$ and let m_a and m_b be a pair of black pixels on $\mathcal{M}$ through which the* cut *pixels on this part of the boundary are connected to $\mathcal{M}$. Now, m_a and m_b are connected only through this part of the boundary. Moreover, this is true for every such part of the boundary.*

Proof. Let us suppose that we have another path connecting m_a and m_b. This clearly implies that there is a *non-cut* pixel on the part of the boundary contradicting the hypothesis. The same argument follows for all such parts of the boundary. □

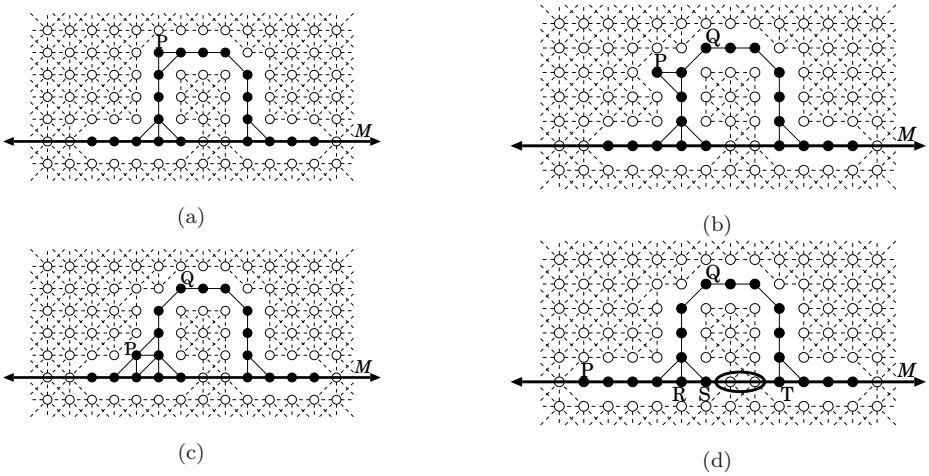


Fig. 2. (a) This shows a connected component in a layer. P is a *non-cut* pixel and M, the *Merge Axis*. (b) This shows P in its new position after the interchange with its adjacent white pixel. (c) P is interchanged with white pixels twice more. (d) P is finally placed on M. This leaves all other pixels on the boundary to be *cut* pixels. Q is one such pixel. The oval region shows the disconnectivity on the *Merge Axis* before collapsing the *cut* pixels.

For an illustration, Fig. 2(d) shows a discontinuity on $\mathcal{M}$ with all the black pixels on the boundary and not on $\mathcal{M}$ being *cut* pixels. The pixels S and T are connected only through this boundary.

So, our goal is to fill the gaps between the two ends on $\mathcal{M}$. Consider the leftmost pixel in the topmost level, say P. As this is the topmost level, this pixel has no $\mathbf{B}_8$ neighbors in the level above or to the left of it. Now, considering the remaining possibilities the only two situations where there are no *non-cut* pixels on the boundary are illustrated in Fig. 3. As it is clear from the figure, filling up of the gaps on $\mathcal{M}$ can be clearly done by collapsing P to the *level* below it.

Figure 3(c) shows the collapsing of Q for example. Q is interchanged with the pixel right below it. This is formed from Fig. 2(d). It is easy to see that collapsing preserves connectivity.

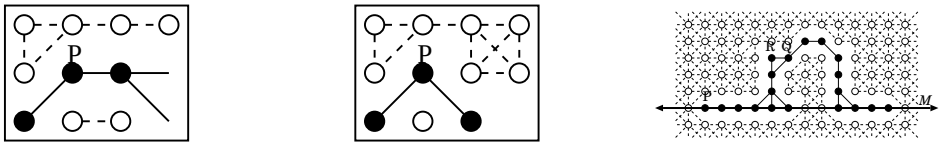


Fig. 3. (a) One possible location of P, the leftmost pixel on the topmost level. (b) The other possible location of P. (c) The *cut* pixel Q is collapsed onto the previous *level*, exposing a *non-cut* pixel R.

We continue collapsing. If this results in a new *non-cut* pixel, we go for the next iteration.

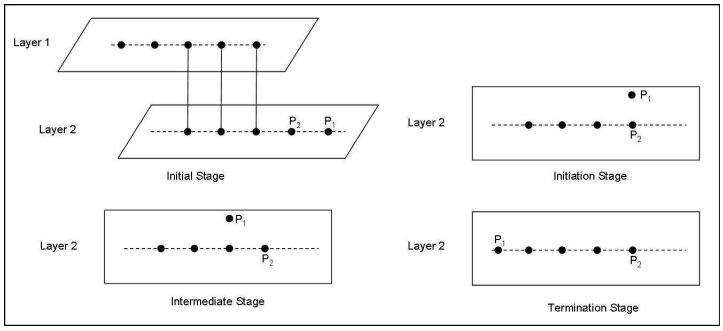


Fig. 4. Here P_1 and P_2 needed to be displaced. In this illustration, the transformation of P_1 is shown. P_1 is moved along the chain (for preserving the connectivity) and brought back to the chain whenever the first white pixel is found. The displacement of P_2 can be similarly done.

Complexity Analysis: Assume that the total number of black pixels in the 2D region is n. Any pixel can be a *cut* or a *non-cut* pixel at any point of time during the transformation. If it is a *non-cut* pixel, and if it is chosen to be moved along the boundary to $\mathcal{M}$, it takes $O(n)$ interchanges to reach $\mathcal{M}$, as the boundary contains at most n pixels. If it is a *cut* pixel, all pixels other than those on $\mathcal{M}$ are *cut* pixels and this particular pixel has been chosen to be collapsed to a *level* below it, then it takes one interchange to do so. There can be at most n such interchanges for any particular pixel. Hence, for any pixel, it takes at most $O(n)$ interchanges and therefore, the complexity is $O(n^2)$.

3.2 Algorithm-Part I

Let u be a node with degree one in G' and also let (u, v) be the edge emerging from u. In other words, the components represented by u and v, say U and V respectively, have at least one $\mathbf{B}_{26}$ adjacent voxel pair (v_u, v_v), with $v_u \in U$ and $v_v \in V$. As the degree of u in G' is one, we develop a strategy to *merge* U with V.

To make the merging easier, we first form a single straight chain of all the black pixels on U. The merge axis $\mathcal{M}$ for U can be in any direction. So, let us fix it to be horizontal. We merge all the black pixels in U on $\mathcal{M}$.

Let us suppose that U and V are in a layers i and $i+1$ $(i-1)$, respectively. Again, to make merging easier we take $\mathcal{M}$ to such a location on layer i that the top view of these two components U and V looks like, $\mathcal{M}$ protruding out from the boundary of V. So, we translate $\mathcal{M}$ horizontally, in the layer in which U is present, say i, till any further move removes the connectivity between U and V, using the procedure described below.

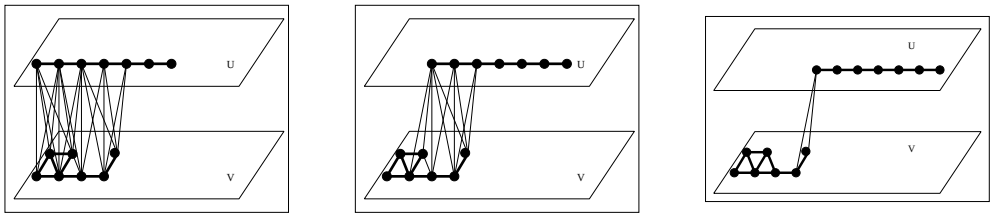
Translation: The idea behind translation is simple. We move pixel by pixel.

Figure 4 shows an example of how we do it. The extreme pixel is moved first followed by the next farthest pixel. A given pixel gets displaced $O(n)$ times. There are $O(n)$ pixels to be moved. Hence, the complexity for translation is $O(n^2)$.

If U intersects with any other connected component in the *layer* i either during the process of merging on $\mathcal{M}$ or during the process of translation, we do the following.

(1) We stop the process.
(2) We consider the compound component formed by U and the component with which it intersects instead of the original components.
(3) We build a new G and form the new spanning tree, G'.
(4) We go for the next iteration.

Note that, U might have established new links with other components in layers $i-1$ and $i+1$. Figure 5 shows an illustration of this part.



(a) (b) (c)

Fig. 5. (a) The pixels in U have been merged on to its *Merge Axis*. This shows all the adjacencies. (b) Two of the pixels have been translated by the procedure described above. (c) The situation after the entire translation.

Complexity (Part I): Let n_u and n_v be the number of black pixels in U and V respectively. From our earlier discussions, merging all n_u pixels on $\mathcal{M}$ takes $O(n_u^2)$ interchanges. As translating $\mathcal{M}$ horizontally by one pixel takes $O(n_u)$ interchanges, the entire translation phase takes $O(n_u n_v)$ interchanges. If any process has to be stopped in the middle, then a new iteration has to be started after making some changes, mentioned above.

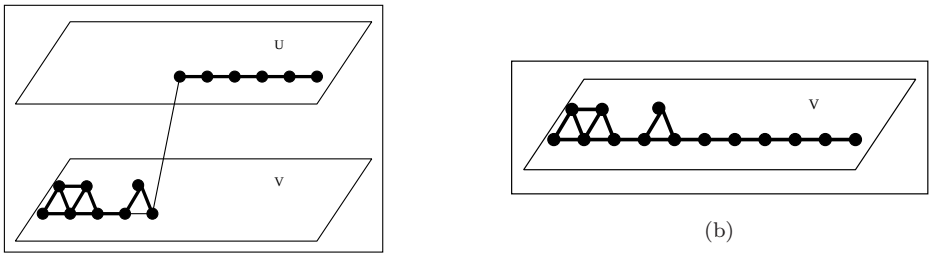
3.3 Algorithm-Part II

Now, we merge the chain in U with the layer containing V. The pixels can be merged in any order but we restrict to one particular order, namely, from the end of the *Merge Axis* on U which is $\mathbf{B}_{26}$ adjacent to a pixel on V to the other end. There are two possibilities.

Case I: U has developed new $\mathbf{B}_{26}$ adjacencies with some voxels of other components in the layer containing V.

Case II: No such adjacency has been developed.

It is very well possible that some other edge between U and any other component W in G got snapped. Case II is the easiest of the two. All we need to do is, keep interchanging the voxels on $\mathcal{M}$ in U, with the white voxels in the layer containing V starting from either end of $\mathcal{M}$. Figure 6 shows an example. It is easy to see that the complexity in this case is $O(n_u)$. Now, let us



(a) (b)

Fig. 6. (a) Starting from the situation in Fig. 5(c) a pixel has been merged with V. (b) All pixels on U have been merged onto V.

consider Case I. Then, there is a possibility that, if we follow the same steps as suggested above for Case II, after certain number of steps, we encounter another connected component. Figure 7 shows an example. If we encounter such a component, we adopt a sequence of steps, similar to those considered in *Part I*.

1. We stop the process. This may be the end of the process.
2. We consider the compound component formed by V and the other component in the same layer which we encountered, along with the pixels interchanged between U and this layer by now, instead of V and that other component.
3. We update U. U now contains fewer pixels on the chain $\mathcal{M}$.
4. We rebuild G and form the new spanning tree, G'.
5. We start a new iteration.

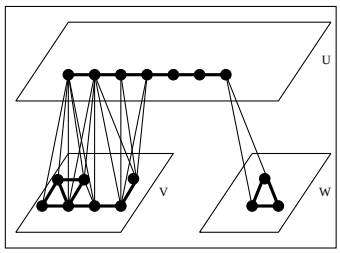


Fig. 7. U develops new adjacencies with W, during translation.

Lemma 2 *The algorithm suggested above, eventually leads us to the intermediate structure, a single chain containing all the black voxels in the original image I.*

Proof. In one pass through *Part I* of the algorithm, we either merge U with another connected component in the same layer, i, or move the chain, $\mathcal{M}$ to a new location, again in the same layer, i. So, decrease in $|V_G|$ in *Part I* is less than or equal to one. In one pass through *Part II* of the algorithm, we merge either U with V or V with some other connected component in its layer or both. So, decrease in $|V_G|$ is either one or two.

Now, if any pass through *Part I* merges two connected components, we don't touch *Part II* until again we pass through *Part I*, as a new iteration is started. If the pass through *Part I* doesn't merge but, simply translates U to a new location, we definitely pass through *Part II* and this guarantees that at least two components will be merged. Hence, each iteration through the algorithm reduces $|V_G|$ by at least one and therefore, after at most $|V_G| - 1$ iterations, we are left with a single component. Now, we can form $\mathcal{M}$ for this single component in any direction starting from anywhere and form the single chain. □

Let us find the complexity of an iteration. Assume that *component i* has n_i number of black pixels and that there are c components in total. Note that, components in a particular layer may be disconnected but they can be connected using voxels of layers above and below. Let $\sum_{i=1}^{c} n_i = n$. We divide the complexity calculation into two parts as follows.

(1) Merging of all the pixels in a single component.
(2) Merging of different components.

Merging a component of n_i black pixels onto its *Merge Axis* takes $O(n_i^2)$ interchanges. Now, during the process, if this intersects with another component with n_j number of black pixels, we simply start a new iteration. Let n_k be the total number of pixels of the component on which this *Merge Axis* has to be merged. Translation of the *Merge Axis* takes $O(n_i n_k)$ interchanges if it doesn't intersect with any other component. Else, we simply start a new iteration. Once translation is done, merging takes $O(n_i)$ interchanges if it's Case II. In Case I, we have to start a new iteration somewhere in the middle.

So, the worst case complexity of an iteration is

$$O(n_i^2) + O(n_i n_k) + O(n_i) = O(n_i^2 + n_i n_k)$$

And the overall worst case complexity is simply a summation of the above complexity over all the iterations. From Lemma 2 it is clear that the number of iterations is at most $c-1$. Note that n_i, n_k may change after every iteration due to merging of components. In any case, n_i and n_k are $O(n)$. So, an upper bound of the complexity is

$$\Sigma_{i,k=1}^{c-1} O(n_i^2 + n_i n_k) = \Sigma_1^{c-1} O(n^2) = O(cn^2)$$

Theorem 1 *Let I and J be two $\mathbf{B}_{26}$-connected 3-dimensional images each having n black voxels with the total number of 8-connected components in all 2-dimensional layers of I and J being c_1 and c_2 respectively. I and J are transformable and I can be converted to J using a sequence of $O((c_1 + c_2)n^2)$ 26-local interchanges.*

Proof. The theorem follows from Lemma 2 and the above discussion. □

4 Voxel Transformation in $\mathbf{B}_{26}$ model under Single Backbone Condition

In the model presented till now, a *valid interchange* is taken as such an interchange between any two $\mathbf{B}_{26}$ adjacent black and white voxels, which preserves the connectivity of the image *before* and *after* the interchange. A slightly different model can be obtained if we impose a *single backbone* condition [2] along with our original connectivity model. Dumitrescu and Pach [3] also consider this as an alternative model. In our case, a *backbone* is defined as the set of all black voxels except the one which we currently interchange. The condition is that the *backbone* must be $\mathbf{B}_{26}$ connected during and after each interchange. In order to adopt this model, we only need to make small changes in our algorithm.

First, note that, in the algorithm we described, there are only two situations where the *single backbone* condition fails.

(1) While collapsing the pixels on the boundary when all the *non-cut* pixels not on *Merge Axis* are exhausted to form the 2D linearly connected chains.
(2) While merging the translated *Merge Axis* with a component in an adjacent layer.

First Situation: While forming the 2D linearly connected chains, instead of collapsing the pixels to the *level* below when all the pixels on the boundary are *cut*, we can move the black pixels between m_a and m_b on the *Merge Axis* (refer Lemma 1) to one of the extreme ends of the axis (through interchanges). This is similar to the *Translation*, mentioned in *Part I* in section 3.2. So, ultimately what we have is a *Merge Path* which is the union of two parts of the original

Merge Axis and the *cut* black pixels on the boundary. For example, consider Fig. 2(d). The pixels R, S and T have to be translated to the ends of M.

This can be easily adopted to the algorithm discussed. We need to consider only one *connectivity-sensitive pixel* for each 2D connected component. So, we can easily decide which part of the *Merge Axis* is to be extended and which part should be left untouched (depending on which part the *connectivity-sensitive pixel* lies on). For example, suppose that in Fig. 2(d), the pixel R is the *connectivity-sensitive pixel*. We should not move R during the translation mentioned above. So, a possible and easy solution is to translate all the pixels starting from the rightmost end of M till T to the left end of M. Then, translate S. And we end up with the *Merge Path*.

Now, we are left with *bending* the *Merge Path* to a straight line. The only curvy portion is that of the chain of black pixels. Again, in a similar manner, considering the above example, translate each pixel on the chain, starting from the right end to the left end of M.

Second Situation: We only need to change the order in which the pixels on the *Merge Axis* are merged with the component in the other layer. Note that the end of the *Merge Axis* other than the one whose removal disconnects the components, is a *non-cut* pixel. So, we can merge starting from that end, just the opposite way we mentioned in Section 3.3. Now, clearly, interchanging with a *non-cut* pixel maintains the backbone's connectivity. The only problem is with Case I of the Section 3.3. The new adjacencies are at the very end where we have *non-cut* pixels. One possible solution is starting from this end, find the first pixel which is not $\mathbf{B}_{26}$ adjacent with any of the pixels in the new component which the Case I refers to. So, starting from this pixel (which is clearly *non-cut*) keep merging till the other end. The rest of the algorithm follows.

The changes mentioned in both the above mentioned situations do not change the complexity.

5 $\mathbf{B}_{26}$,$\mathbf{W}_{26}$-Connectivity Preserving Transformation

Consider a right hand Cartesian system. Let $v = (x, y, z)$ be a voxel such that

(1) v is black.
(2) $(x, y-1, z-1)$, $(x, y, z-1)$, $(x, y+1, z-1)$ are all white.
(3) There exists an integer $k \geq 0$ such that for every integer δ with $1 \leq \delta \leq k$, $(x, Y, z+\delta)$ is black for some Y (may be different for different values of δ) with the condition that every such $(x, Y, z+\delta)$ has $(x, Y', z+\delta-1)$ as

its $\mathbf{B}_{26}$ neighbor with $Y' = y$ for $\delta = 1$ and all voxels $(x, Y, z+k')$, $k' > k$ for any Y are white.

(4) All voxels (X, Y, Z) with $X \geq x+1$, $Z \geq z-1$ are white.

(5) z is maximum (of all the voxels satisfying the above conditions).

(6) $x > 0$

Given that the last condition is satisfied by some voxel, a voxel satisfying the remaining conditions always exists. Consider the Y-Z plane passing through the maximum x-coordinate of any black voxel. Take a black voxel v' with the maximum z-coordinate in this plane. Consider a maximal $\mathbf{B}_8$ connected monotonically descending path (descending in z direction) in this plane. Let the path be $v' \rightsquigarrow v''$. As the path is maximal, the second condition is satisfied by v''. As this plane has the maximum x-coordinate the fourth condition is also satisfied. The third condition is obviously satisfied because we started off with the voxel with maximum z-coordinate. Now, there can be many such v'' (for many such v' or otherwise). Take the one with maximum z-coordinate as v (the fifth condition). The algorithm exits when such a voxel cannot be found, which implies that every voxel has the same first coordinate (which is 0) and we have a 2-D image.

In the following, the $\mathbf{B}_{26}$,$\mathbf{W}_{26}$-connectivity preserving transformation is developed using two exhaustive cases. Note that in any interchange we describe below the white voxel with which v is interchanged never belongs to the set of white voxels $\{(x, y-1, z-1), (x, y, z-1), (x, y+1, z-1)\}$ and hence, is not isolated after the interchange as (x, y, z) is a 26-neighbor of the voxels in the above set. Thus, whenever we discuss about the connectivity of the white in any image resulting after an interchange, we concentrate only on the white voxels other than the one with which v is interchanged.

Case 1: v is not a cut vertex of $\mathbf{B}_{26}(I)$. Consider the five voxels $(x, y+1, z)$, $(x, y+1, z+1)$, $(x, y, z+1)$, $(x, y-1, z+1)$ and $(x, y-1, z)$. If any of them is black and y' is the y-coordinate of that voxel, it is easy to see that $(x+1, y', z+1)$ is not a cut vertex of $\mathbf{W}_{26}(I)$ ($G[A_{W_{26}}((x+1, y', z+1))]$ is connected by the choice of v) and hence, the interchange $\langle (x, y, z), (x+1, y', z+1) \rangle$ preserves the connectivity by Observation 2. Figure 8(a) shows the situation where all the above five voxels are black.

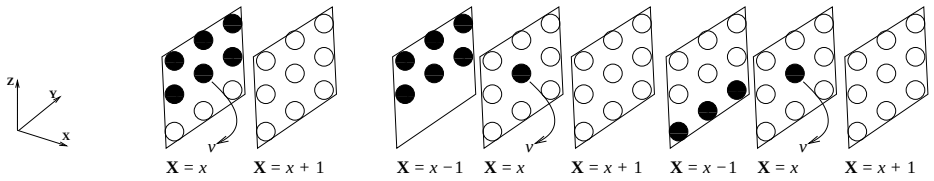


Fig. 8. Illustration of the possible situations in Case 1 of the algorithm for $\mathbf{B}_{26}, \mathbf{W}_{26}$ transformation.

If all the above five voxels are white, consider the six voxels $(x-1, y+1, z)$,

$(x-1, y, z)$, $(x-1, y-1, z)$, $(x-1, y+1, z+1)$, $(x-1, y, z+1)$ and $(x-1, y-1, z+1)$. Now, by the choice of v, all voxels $(x, Y, z+k)$ for any integer $k > 0$ and for any value of Y are white. If any of the above six voxels is black and y' is the y-coordinate of that voxel we have that $(x, y', z+1)$ is not a cut vertex of $\mathbf{W}_{26}(I)$ as again $G[A_{W_{26}}((x, y', z+1))]$ is connected (by Observation 1). So, by Observation 2 the interchange $\langle (x, y, z), (x, y', z+1) \rangle$ preserves the connectivity. Figure 8(b) shows the situation where all the above six voxels are black.

In the case that all the above six voxels are also white, then at least one of the three voxels $(x-1, y+1, z-1)$, $(x-1, y, z-1)$ and $(x-1, y-1, z-1)$ must be black for the image is connected all other $\mathbf{B}_{26}$ neighbors of v are white. Then, we have that $G[A_{W_{26}}((x-1, y, z))]$ is connected and by Observation 1 $(x-1, y, z)$ is not a cut vertex of $\mathbf{W}_{26}(I)$. Hence, the interchange $\langle (x, y, z), (x-1, y, z) \rangle$ preserves the connectivity. Figure 8(c) shows the situation where all the above three voxels are black.

Case 2: v is a cut vertex of $\mathbf{B}_{26}(I)$. In this case, $(x-1, y, z)$ has to be white otherwise $A_{B_{26}}(v) \subseteq A_{B_{26}}[(x-1, y, z)]$ and by Observation 1, v will not be a cut vertex of $\mathbf{B}_{26}(I)$ (Figure 9(a)). Consider the three voxels $(x, y, z+1)$, $(x-1, y, z+1)$ and $(x-1, y, z-1)$ (shown as all black in Figure 9(b)). Depending on whether each of these voxels is black or white, we have several sub-cases.

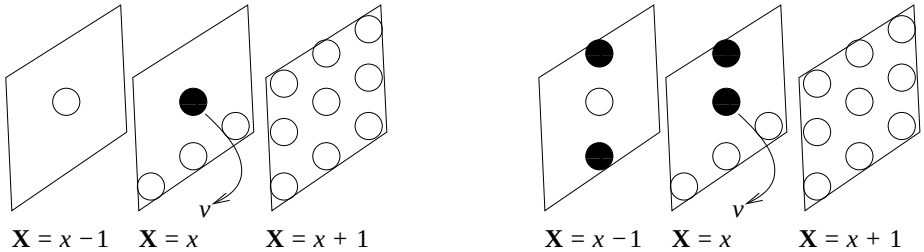


Fig. 9. Illustration of Case 2 of the algorithm for $\mathbf{B}_{26}, \mathbf{W}_{26}$ transformation.

Let $(x-1, y, z-1)$ and $(x-1, y, z+1)$ both be black (Figure 10(a)). $(x, y, z+1)$ can be either black or white. Observe that $A_{B_{26}}(v) \subseteq A_{B_{26}}((x-1, y, z-1)) \cup A_{B_{26}}((x-1, y, z+1))$. If $(x-2, y, z)$ is black, it is also the black neighbor of $(x-1, y, z-1)$ and $(x-1, y, z+1)$ and hence, $G[A_{B_{26}}(v) \cup \{(x-2, y, z)\}]$ is connected and v is non-cut (Observation 1). So, $(x-2, y, z)$ is white. Now, the interchange $\langle (x, y, z), (x-1, y, z) \rangle$ preserves the connectivity of the black in the resulting image as $A_{B_{26}}(v) \subseteq A_{B_{26}}[(x-1, y, z)]$. If $(x-1, y, z)$ is not a cut vertex of $\mathbf{W}_{26}(I)$, then we can do the above interchange. Else, still consider the above interchange and the resulting image I'. If the white is disconnected in I', that is only because we have a *closed black surface* through $(x-1, y, z)$ enclosing a white component. This implies that there is a B_4 cycle in all 2-D planes through $(x-1, y, z)$, in particular in the planes, $X-Y$, $Y-Z$ and $X-Z$. But in the plane $X-Z$, such a cycle implies an alternative path between $(x-1, y, z-1)$ and $(x-1, y, z+1)$ not

through v (in I) making v non-cut. Thus, white is also connected in I' and the interchange preserves the connectivity.

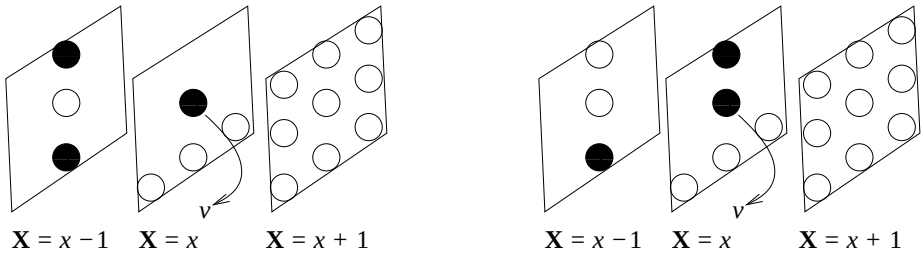


Fig. 10. Two of the possible sub-cases in Case 2 of the algorithm for $\mathbf{B}_{26}, \mathbf{W}_{26}$ transformation.

Now, let $(x-1, y, z-1)$ and $(x, y, z+1)$ be black while $(x-1, y, z+1)$ is white (Figure 10(b)). Observe that $A_{B_{26}}(v) \subseteq A_{B_{26}}((x-1, y, z-1)) \cup A_{B_{26}}((x, y, z+1))$. Consider the interchange $\langle (x, y, z), (x-1, y, z) \rangle$. This preserves the connectivity of the black in the resulting image. If the white in the resulting image I' is disconnected (otherwise, we do the above interchange), there is a B_4 cycle through $(x-1, y, z)$ in the plane $X-Z$ which implies $(x-2, y, z)$ is black. Moreover, there is a path between $(x-2, y, z)$ and $(x-1, y, z-1)$ in I (not through v). Now, if $(x-1, y, z+1)$ is non-cut, the interchange $\langle (x, y, z), (x-1, y, z+1) \rangle$ preserves the connectivity as $(x-1, y, z-1)$ and $(x, y, z+1)$ are connected in I' through $(x-1, y, z+1)$ and $(x-2, y, z)$. So, assume it is cut. If $(x-2, y, z+1)$ is white, $G[A_{W_{26}}((x-1, y, z+1)) \cup \{(x, y, z-1), (x+1, y, z), (x+1, y, z+1)\}]$ is connected and $(x-1, y, z+1)$ is not cut (Observation 1). So, $(x-2, y, z+1)$ is black. If $(x-1, y, z+2)$ is black, $G[A_{B_{26}}(v) \cup \{(x-1, y, z+2), (x-2, y, z), (x-2, y, z+1)\}]$ is connected and v is non-cut (Observation 1). So, $(x-1, y, z+2)$ is white and after the interchange $\langle (x, y, z), (x-1, y, z+1) \rangle$, the resulting image I' has a disconnected white only if there is a B_4 cycle through $(x-1, y, z+1)$ in the plane $X-Z$. This implies that there is a path between $(x, y, z+1)$ and $(x-2, y, z+1)$ in I' and hence, a path between them and between $(x, y, z+1)$ and $(x-1, y, z-1)$ in I not through v again implying v is non-cut in I. Thus, the white is also connected in I' and the interchange preserves the connectivity.

Consider the case where $(x-1, y, z-1)$ is black and both $(x, y, z+1)$ and $(x-1, y, z+1)$ are white. If the interchange $\langle (x, y, z), (x-1, y, z) \rangle$ does not disconnect the white in the resulting image, it also preserves the connectivity. Else, we should have a B_4 cycle in each of the $X-Y$, $Y-Z$ and $X-Z$ planes through $(x-1, y, z)$ in the resulting image. In the $Y-Z$ plane with x-coordinate '$x-1$', this implies that at least one of $(x-1, y+1, z)$ and $(x-1, y-1, z)$ is black. Let $(x-1, y-1, z)$ be black. (The other case can be argued for similarly.) This situation is shown in Figure 11(a). As shown in the figure, we cannot have $(x, y+1, z)$ or $(x-1, y+1, z)$ to be black as otherwise, $G[A_{B_{26}}(v)]$ will be connected and by Observation 1 v will not be

a cut vertex of $\mathbf{B}_{26}(I)$. Consider the interchange $\langle (x,y,z),(x,y,z+1)\rangle$ and the $\mathbf{W}_{26}$-neighbors of the voxel $(x,y,z+1)$. All the nine 26-neighbors in the plane $X = x+1$ are white and hence belong to $\mathbf{W}_{26}$. Also, $(x-1,y,z+1)$ is white (as argued in the beginning of this paragraph). This is sufficient to conclude that $G[A_{W_{26}}((x,y,z+1))]$ is connected and hence, $(x,y,z+1)$ is not a cut vertex of $\mathbf{W}_{26}$ (Observation 1). Also, $A_{B_{26}}(v) \subseteq A_{B_{26}}((x-1,y,z-1)) \cup A_{B_{26}}((x,y,z+1))$ and in the resulting image after this interchange, $(x,y,z+1)$ and $(x-1,y,z-1)$ are connected through $(x-1,y-1,z)$ making the black connected too. Thus, the above interchange preserves the connectivity.

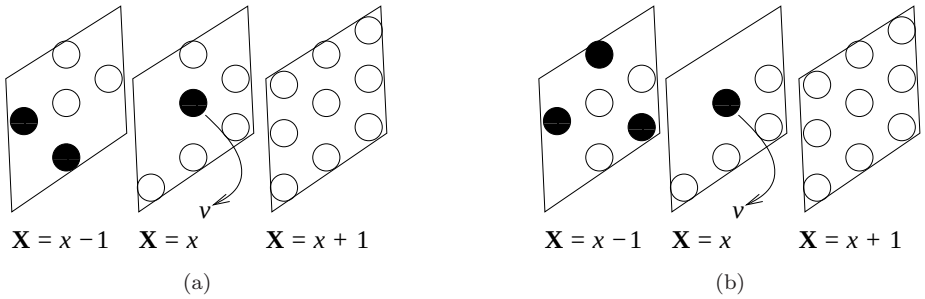


Fig. 11. Two more possible sub-cases in Case 2 of the algorithm for $\mathbf{B}_{26}, \mathbf{W}_{26}$ transformation.

Now, consider that $(x-1,y,z-1)$ is white. If $(x-1,y,z+1)$ is white, there can be no B_4 cycle through $(x-1,y,z)$ (after an interchange of v with that white voxel) in the plane $Y = y$ and hence, the interchange $\langle (x,y,z),(x-1,y,z)\rangle$ preserves the connectivity (it is easy to see that the black is also connected). So, we assume that $(x-1,y,z+1)$ is black and that the interchange $\langle (x,y,z),(x-1,y,z)\rangle$ makes the white disconnected in the resulting image (otherwise, we do this interchange). So, there have to be three B_4 cycles in the three 2-D planes passing through $(x-1,y,z)$ (after interchanging it with v) and hence, at least one of $(x-1,y-1,z)$ and $(x-1,y+1,z)$ is black. Assume that $(x-1,y-1,z)$ is black (the other case can be similarly argued for). This implies that both $(x-1,y+1,z)$ and $(x,y+1,z)$ are white as otherwise $G[A_{B_{26}}(v)]$ will be connected and by Observation 1 v will not be a cut vertex of $\mathbf{B}_{26}(I)$. For the same reason, $(x-1,y+1,z-1)$ has to be black. This situation is shown in Figure 11(b). If $(x-2,y,z)$ is black, it is easy to see that $G[A_{B_{26}}(v) \cup \{(x-2,y,z)\}]$ will be connected and by Observation 1 v will not be a cut vertex of $\mathbf{B}_{26}(I)$. If it is white, there can be no B_4 cycle through $(x-1,y,z)$ after the interchange $\langle (x,y,z),(x-1,y,z)\rangle$ in the plane $Y = y$ making the white in the resulting image connected contradicting the assumption. Thus, the above interchange preserves the connectivity.

In all the above cases, $\phi(I)$ decreases after every interchange.

From the development of the previous transformation, we obtain as a conclusion the following theorem.

Theorem 2 *Any two* $\mathbf{B}_{26}$,$\mathbf{W}_{26}$*-connected images* I *and* J *each having* n *black voxels are transformable and* I *can be converted to* J *using a sequence of* $O(n^2)$ 26*-local interchanges.*

6 The General Case: $\mathbf{B}_{3^N-1}$,$\mathbf{W}_{3^N-1}$ Connectivity Preserving Transformation

Lemma 3 *Any* $\mathbf{B}_{3^N-1}$,$\mathbf{W}_{3^N-1}$*-connected* N*-dimensional image* I *of* n *black pixels can be converted to some* $\mathbf{B}_{3^{N-1}-1}$,$\mathbf{W}_{3^{N-1}-1}$*-connected* $(N-1)$*-dimensional image* J *also of* n *black pixels such that every intermediate image resulting after an interchange is* $\mathbf{B}_{3^N-1}$,$\mathbf{W}_{3^N-1}$*-connected and the conversion can be done using a sequence of* $O(n^2)$ (3^N-1)*-local interchanges, where* $N > 1$*. (Note that an* $\mathbf{B}_{3^{N-1}-1}$,$\mathbf{W}_{3^{N-1}-1}$*-connected image is also* $\mathbf{B}_{3^N-1}$,$\mathbf{W}_{3^N-1}$*-connected.)*

Proof. Consider the image I. We define a boolean function, $next : \mathbb{Z}^N \rightarrow \{true, false\}$ as $next(\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_N))$ is $true$ iff α is black and there exists a black pixel $\beta = (\beta_1, \beta_2, \ldots, \beta_N)$ such that $\beta_1 = \alpha_1$, $\beta_2 = \alpha_2 + 1$ and $\beta_i \in \{\alpha_i - 1, \alpha_i, \alpha_i + 1\}$ for $i \geq 3$. Also define $succ : \mathbb{Z}^N \rightarrow \mathbb{Z}^N$ as $succ(\alpha) = \beta$ if α is black and $next(\alpha) = true$ with β as in the definition of $next()$ ($succ()$ is undefined otherwise).

Let $p = (\delta_1, \delta_2, \ldots, \delta_N)$ be a pixel such that

(1) p is black.
(2) All pixels $(d_1, d_2, \ldots, d_N)$ with $d_1 = \delta_1$, $d_2 = \delta_2 - 1$ and $d_i \in \{\delta_i - 1, \delta_i, \delta_i + 1\}, i \geq 3$ are white.
(3) There exists a sequence of black pixels $(p = p_1, p_2, \ldots, p_k)$ with $next(p_i) = true$, $succ(p_i) = p_{i+1}$ for every $1 \leq i < k$ and $next(p_k) = false$ such that every pixel $(d_1, d_2, \ldots, d_N)$ with $d_1 = \delta_1$, $d_2 > \delta_2 + k$ is white.
(4) All pixels $(d_1, d_2, \ldots, d_N)$ with $d_1 \geq \delta_1 + 1$ and $d_2 \geq \delta_2 - 1$ are white.
(5) δ_2 is maximum.
(6) $\delta_1 > 0$.

Given that the last condition is satisfied by some pixel, a pixel satisfying conditions 1 through 4 can always be found among the set of all black pixels with the maximum value of the first coordinate. The fifth condition is to select one among many such pixels. The sixth condition ensures that ϕ never crosses zero as we use this particular pixel for the interchange. The algorithm exits when such a pixel cannot be found, which implies every pixel has the same first coordinate (which is 0).

We now define two useful functions. Consider the set of all sequences of length between 1 and N (inclusive) of the integers -1, 0 and 1, $\mathbf{S}$. Define $s : \mathbb{Z}^N \times \mathbf{S} \rightarrow \mathbb{Z}^N$ as $s(((\alpha_1,\alpha_2,\ldots,\alpha_N),(S_1,S_2,\ldots,S_k))) = (\alpha_1 + S_1,\alpha_2 + S_2,\ldots,\alpha_k + S_k,\alpha_{k+1},\alpha_{k+2},\ldots,\alpha_N)$. Also, define $g : \mathbb{Z}^N \times \mathbf{S} \rightarrow \mathcal{P}(\mathbb{Z}^N)$ as $g(((\alpha_1,\alpha_2,\ldots,\alpha_N),(S_1,S_2,\ldots,S_k))) = \{(\beta_1,\beta_2,\ldots,\beta_N) \mid \beta_i = \alpha_i + S_i \text{ for } 1 \leq i \leq k \text{ and } \beta_i \in \{\alpha_i - 1,\alpha_i,\alpha_i + 1\} \text{ for } i > k\}$ ($\mathcal{P}(\mathbf{A})$ is the power set of $\mathbf{A}$). We use the notation $s_\alpha(S_1,S_2,\ldots,S_k)$ and $g_\alpha(S_1,S_2,\ldots,S_k)$ for brevity. Clearly, for a pixel α, $s_\alpha(t_1,t_2,\ldots,t_l) \in A_{\mathbf{G}_{3^N-1}}[\alpha]$ and $g_\alpha(t_1,t_2,\ldots,t_l) \subset A_{\mathbf{G}_{3^N-1}}[\alpha]$ for some $(t = (t_1,t_2,\ldots,t_l)) \in \mathbf{S}$. Also, it is easy to see that $g_\alpha(t_1,t_2,\ldots,t_m,-1) \cup g_\alpha(t_1,t_2,\ldots,t_m,0) \cup g_\alpha(t_1,t_2,\ldots,t_m,1) = g_\alpha(t_1,t_2,\ldots,t_m)$ for any $(t = (t_1, t_2, \ldots, t_m)) \in \mathbf{S}$ with $0 < m < N$ and $g_\alpha(-1) \cup g_\alpha(0) \cup g_\alpha(1) = A_{\mathbf{G}_{3^N-1}}[\alpha]$.

We do the following case analysis. We use $(d_1, d_2, \ldots, d_N)$ to mean an arbitrary pixel in $\mathbb{Z}^N$ and $d_i(\alpha)$ to mean the value of the i^{th} coordinate of a pixel α (we could have as well defined N functions for the same). Note that in any interchange we describe below the white pixel with which p is interchanged never belongs to the set of white pixels $g_p(0,-1)$ and hence, is not isolated after the interchange as p is a (3^N-1)-neighbor of the pixels in the above set. Thus, whenever we discuss about the connectivity of the white in any image resulting after an interchange, we concentrate only on the white pixels other than the one with which p is interchanged.

Case 1 : p is a non-cut vertex of $\mathbf{B}_{3^N-1}(I)$ We know from the choice of p that all the pixels in $g_p(1)$ and $g_p(0,-1)$ are white. If at least one pixel $q \in g_p(0,0) \cup g_p(0,1)$ other than p is black, then consider the interchange $\langle p, r\rangle$ where $r = (\delta_1 + 1, \delta_2 + 1, d_3(q), \ldots, d_N(q))$. It is easy to see that $r \in g_q(1,0) \cup g_q(1,1)$ (to be specific, r is either $s_q(1,0)$ or $s_q(1,1)$) and hence, the black is connected in the resulting image. Also, $A_{W_{3^N-1}}(r) \subseteq A_{W_{3^N-1}}(s_r(0,1)) \cup A_{W_{3^N-1}}(s_r(0,-1))$ and $s_r(0,-1)$ is connected to $s_r(0,1)$ through $s_r(1)$ which imply $G[A_{W_{3^N-1}}(r)]$ is connected and hence, r is non-cut in $\mathbf{W}_{3^N-1}(I)$. Thus, the interchange preserves the connectivity.

Else, if at least one pixel $q \in g_p(-1,0) \cup g_p(-1,1)$ is black, consider the interchange $\langle p, r\rangle$ where $r = (\delta_1, \delta_2 + 1, d_3(q), \ldots, d_N(q))$. Now that $next(p) = false$ in the original image, we have that $G[A_{W_{3^N-1}}(r) \cup \{s_p(0,-1)\}]$ is connected and hence, r is non-cut in $\mathbf{W}_{3^N-1}(I)$. Also, the black is connected in the resulting image. Thus, the interchange preserves the connectivity.

Else, at least one pixel $q \in g_p(-1,-1)$ has to be black as otherwise p will be isolated. Consider the interchange $\langle p, r\rangle$ where $r = s_p(-1,1)$. For $N > 2$, it is easy to see that $A_{W_{3^N-1}}(r) \subseteq A_{W_{3^N-1}}(s_r(0,0,1)) \cup A_{W_{3^N-1}}(s_r(0,0,-1))$ and $s_r(0,0,1)$ and $s_r(0,0,-1)$ are connected through $s_r(1)$ which imply $G[A_{W_{3^N-1}}(r)]$ is connected and hence, r is non-cut in $\mathbf{W}_{3^N-1}(I)$. For $N = 2$, by the choice of p (especially the condition 5), $s_r(0,1)$ has to be white. Note that $s_r(0,1)$ satisfies all the first four conditions for the selection of p. Also, this need not be the case for $N > 2$. (We cannot make this deduction if

$\delta_1 = 1$, but in that case r is an obvious non-cut vertex in $\mathbf{W}_{3^N-1}(I)$.) So, following similar arguments as in the previous paragraphs, r is again non-cut. So, as long as the black is connected in the resulting image, we can do this interchange. Now, consider the pixel $q' = s_{s_p(-1)}(-1)$. If this is black in the original image, this is also the neighbor of r in the resulting image due to the above interchange and hence the black is connected. Thus, if the black is not connected in the resulting image, we conclude that q' is white in the original image. Now, $G[A_{W_{3^N-1}}(s_p(-1)) \cup \{s_p(1)\}]$ is connected and hence, $s_p(-1)$ is non-cut. Thus, the interchange $\langle p, s_p(-1)\rangle$ preserves the connectivity.

Case 2 **:** p **is a cut-vertex of** $\mathbf{B}_{3^N-1}(I)$ $s_p(-1)$ has to be white as otherwise $A_{B_{3^N-1}}(p) \subseteq A_{B_{3^N-1}}(s_p(-1))$ and hence, p will not be cut. Consider the three pixels $p_1 = s_p(-1,-1)$, $p_2 = s_p(-1,1)$ and $p_3 = s_p(0,1)$. We have eight cases depending on whether each of these pixels is white or black. We cover all these eight cases in the following.

Case 2(a) **:** p_1 **and** p_2 **are black** Note that p_3 can be white or black. So, we aim to cover two of the eight cases here. Observe that $A_{B_{3^N-1}}(p) \subseteq A_{B_{3^N-1}}(p_1) \cup A_{B_{3^N-1}}(p_2)$. If $p_4 = s_{s_p(-1)}(-1)$ is black, this is also a neighbor to p_1 and p_2 and hence, $G[A_{B_{3^N-1}}(p) \cup \{p_4\}]$ is connected and p is not cut. So, p_4 is white. Let $p_5 = s_p(-1)$. Now, the interchange $\langle p, p_5\rangle$ preserves the connectivity of the black in the resulting image as $A_{B_{3^N-1}}(p) \subseteq A_{B_{3^N-1}}(p_5)$. Assume p_5 is a cut vertex of $\mathbf{W}_{3^N-1}(I)$ (otherwise, we have the above interchange). Consider the above interchange. If the white gets disconnected after the interchange, it is only because there is a closed black surface through p_5 (p_5 is now black) enclosing a white component. This implies there is a B_4 cycle through p_5 in every 2D plane passing through p_5. (If we split a closed ball along any 2D plane, we get a cycle in that plane. The smallest cycle possible is a single point in the case where the plane is tangential. In our case, p_5 would be that single point (or pixel). But this implies that p_5 being black is redundant for the ball to exist which contradicts the assumption that the white is connected in the original image.) In the 2D plane $D_1 - D_2$ through p_5 in the resulting image, p_5 has only two B_4 neighbors, p_1 and p_2 (the other two are white). So, a B_4 cycle through p_5 implies the existence of an alternative path between p_1 and p_2, not through p, in the original image which makes p non-cut. So, the white is also connected in the resulting image and the interchange preserves the connectivity.

Case 2(b) **:** p_1 **and** p_3 **are black,** p_2 **is white** We aim to cover another case of the eight here. Observe that $A_{B_{3^N-1}}(p) \subseteq A_{B_{3^N-1}}(p_1) \cup A_{B_{3^N-1}}(p_3)$. Assume p_5 ($s_p(-1)$) is cut (otherwise, we do the interchange $\langle p, p_5\rangle$ which preserves the connectivity of the black in the resulting image). If the interchange $\langle p, p_5\rangle$ disconnects the white in the resulting image, we have a B_4 cycle in the plane $D_1 - D_2$ through p_5 (as argued in the previous sub-case). Of the four possible B_4 neighbors we know that p_1 is black

and two are white (p in the resulting image and p_2). So, p_4 ($s_{s_p(-1)}(-1)$) has to be black. This implies there is a path between p_4 and p_1, not through p, in the original image. Assume p_2 is cut (otherwise, we do the interchange $\langle p, p_2 \rangle$ which preserves the connectivity of the black in the resulting image as p_1 and p_3 are connected through p_4 and p_2). If $p_6 = s_{p_4}(0,1)$ is white, it is easy to see that all white pixels in $g_{p_2}(-1) \cup g_{p_2}(0)$ are also neighbors of p_6 and all white pixels in $g_{p_2}(1)$ are neighbors of $s_p(1,1)$. Moreover, there is the path p_6,p_5,$s_p(0,-1)$,$s_p(1)$,$s_p(1,1)$ and hence, $G[A_{W_{3^N-1}}(p_2) \cup \{s_p(0,-1), s_p(1), s_p(1,1)\}]$ is connected making p_2 non-cut. So, p_6 is black. If $s_{p_2}(0,1)$ is also black, there is a path between p_3 and p_1 (p_3,$s_{p_2}(0,1)$,p_6,p_4,p_1) making p non-cut. Thus, $s_{p_2}(0,1)$ is white. Consider the interchange $\langle p, p_2 \rangle$. If this disconnects the white in the resulting image, we again have a B_4 cycle through p_2 in the plane $D_1 - D_2$ in the resulting image which implies there is a path between p_6 and p_3 in the resulting image not through p_2 which implies there is a path between p_6 and p_3 and hence a path between p_1 and p_3 in the original image not through p which again implies the neighborhood of p is connected and p is non-cut. So, the interchange does not disconnect the white in the resulting image and hence, preserves the connectivity.

Note : All the cases considered till now are sufficient for $N = 2$ just as discussed in Bose et al. [1].

Case $2(c)$ **:** p_1 **is black,** p_2 **and** p_3 **are white** We aim to cover the fourth case here. Assume that p_5 is cut (otherwise, $\langle p, p_5 \rangle$). If the interchange $\langle p, p_5 \rangle$ disconnects the white in the resulting image, we have a B_4 cycle in the plane $D_2 - D_3$ through p_5. We know two of the four possible B_4 neighbors, namely p_1 which is black and p_2 which is white. So, at least one of the other two is black for the cycle to exist, say $p_7 = s_{p_5}(0,0,-1)$ (the other one is $s_{p_5}(0,0,1)$; that can be taken up similarly and is not discussed). Thus, we assume the cycle goes through p_7. Now, if at least one of $s_p(0,0,1)$ and $s_p(-1,0,1)$ is black, the neighborhood of p is connected and p is non-cut. So, both these pixels are white. This readily implies $G[A_{W_{3^N-1}}(p_3)]$ is connected. Also, $A_{B_{3^N-1}}(p) \subseteq A_{B_{3^N-1}}(p_1) \cup A_{B_{3^N-1}}(p_3)$ and after the interchange $\langle p, p_3 \rangle$, p_3 is connected to p_1 through p_7 making the black connected too. Thus, the above interchange preserves the connectivity.

Case $2(d)$ **:** p_1 **is white** We aim to cover the remaining four cases here. Considering the interchange $\langle p, p_5 \rangle$, if p_2 is white, there can be no B_4 cycle in the plane $D_1 - D_2$ through p_5 (it has exactly one B_4 neighbor) in the resulting image and hence, the interchange preserves the connectivity irrespective of p_5 being cut in the original image (it is easy to see that the black is also connected in the resulting image). So, assume that p_2 is black. Also, assume that p_5 is cut (otherwise, we do the above interchange) and that the white is disconnected in the resulting image after the above interchange. Now, there is a B_4 cycle through p_5 in the plane

$D_2 - D_3$. As described in the above sub-case, assume that p_7 is black which again implies the two pixels $s_p(0,0,1)$ and $s_p(-1,0,1)$ are white. If every pixel in $g_p(-1,-1,1)$ is white, the black neighborhood of p is connected making it non-cut. So, at least one pixel in the above set is black. Consider the pixel p_4. If this is black, we have p_2, p_7 and any black pixel in $g_p(-1,-1,1)$ as neighbors of p_4 implying $G[A_{B_{3^N-1}}(p) \cup \{p_4\}]$ is connected again making p non-cut. So, p_4 is white. This implies there is no B_4 cycle through p_5 in the plane $D_1 - D_2$ after the interchange $\langle p, p_5 \rangle$ (it has exactly one B_4 neighbor) making the white connected in the resulting image contradicting the assumption. Thus, the interchange preserves the connectivity irrespective of p_5 being cut in the original image.

In all the above cases, $\phi(I)$ decreases after every interchange. □

Theorem 3 *Any two* $\mathbf{B}_{3^N-1}$,$\mathbf{W}_{3^N-1}$*-connected* N*-dimensional images* I *and* J *each having* n *black pixels are transformable and* I *can be converted to* J *using a sequence of* $O(Nn^2)$ (3^N-1)*-local interchanges, where* $N > 1$.

Proof. Using a simple induction on N and Lemma 3, we can show that I can be converted to a 1-dimensional image, i.e., a straight line of n black pixels. We repeat this for J and do the steps in reverse from the straight line back to J. Every time a dimension is reduced, we need $O(n^2)$ interchanges. Thus, the total number of interchanges becomes $O(Nn^2)$. It is easy to see that every $(3^{k-1}-1)$-local interchange is also an (3^k-1)-local interchange for any k and hence, the theorem. □

7 Conclusions and Discussions

We show that two N-dimensional images are transformable under different connectivity and interchange models. For any two $\mathbf{B}_{26}$-connected 3-dimensional images I and J each having n black voxels with the total number of 8-connected components in all 2-dimensional layers of I and J being c_1 and c_2 respectively, are transformable using a sequence of $O((c_1+c_2)n^2)$ 26-local interchanges. We also show $\mathbf{B}_{26}$-connectivity of two images under a more difficult interchange model. The general result shows that any two $\mathbf{B}_{3^N-1}$,$\mathbf{W}_{3^N-1}$-connected N-dimensional images each having n black pixels are transformable using a sequence of $O(Nn^2)$ (3^N-1)-local interchanges, where $N > 1$. A $\mathbf{B}_{3^N-1}$,$\mathbf{W}_{3^N-1}$-connected model seems to be the easiest model to consider as the options of (3^N-1)-local interchanges are more. The proofs presented in this work are based on case analysis. Connectivity preserving transformations on other models of connectivity is turning out to be difficult to handle in our current proof framework. This we think would require a rigorous proof. More general neighborhoods that are based on distance functions (e.g.

hyper-rectangular neighborhood) also need to be considered. We leave these for future exploration.

Acknowledgement

A preliminary version of the paper was presented at the 12th International Workshop on Combinatorial Image Analysis, Buffalo, NY, USA, April 2008 [7]. We would also like to thank the anonymous reviewers for their critical comments and constructive suggestions which have helped us in improving the quality of the paper.

References

[1] P. Bose, V. Dujmovic, F. Hurtado, and P. Morin, "Connectivity-Preserving Transformations of Binary Images", *Computer Vision and Image Understanding*, Elsevier, accepted 2007.
doi:10.1016/j.cviu.2007.06.003

[2] A. Dumitrescu, I. Suzuki, and M. Yamashita, "Motion planning for metamorphic systems: feasibility, decidability and distributed reconfiguration", *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, 409-418, 2004.

[3] A. Dumitrescu, and J. Pach, "Pushing squares around", *Graphs and Combinatorics*, vol. 22, no. 1, 37-50, 2006.

[4] R. Klette, and A. Rosenfeld, *Digital Geometry: Geometric Methods for Digital Picture Analysis*, Morgan Kaufman, Elsevier, New Delhi, India, 2005.

[5] A. Rosenfeld, P. K. Saha and A. Nakamura, "Interchangeable pairs of pixels in digital images", *Pattern Recognition*, vol. 35. no. 9, 1853-1865, 2001.

[6] A. Rosenfeld and A. Nakamura, "Two simply connected sets that have the same area are IP-equivalent", *Pattern Recognition*, vol. 34. no. 2, 537-541, 2002.

[7] A. Komuravelli, A. Sinha and A. Bishnu, "Connectivity Preserving Voxel Transformation", *in: V.E. Brimkov, R.P. Barneva, H.A. Hauptman (Eds.), Combinatorial Image Analysis*, Lecture Notes in Computer Science, Vol. 4958, Springer, Berlin, 2008, pp. 1-12.