

The Generation of N Dimensional Shape Primitives

Pieter P. Jonker¹ and Stina Svensson²

¹ Pattern Recognition Group, Faculty of Applied Sciences,
Delft University of Technology, Lorentzweg 1, 2628 CJ Delft, The Netherlands
pieter@ph.tn.tudelft.nl

² Centre for Image Analysis Lägerhyddvägen 3, SE-75237 Uppsala, Sweden
stina@cb.uu.se

Abstract. This paper describes a method to accelerate the generation of shape primitives for N-dimensional images X_N . These shape primitives can be used in conditions for topology preserving erosion or skeletonization in N-dimensional images. The method is based on the possibility to describe primitives for intrinsic dimensions $\tilde{N} = N - 1$ by quadratic equations of the form

$$x_N = \sum_{n=0}^{n=N-1} (a_n x_x + b_n x_n^2).$$

Keywords: N-dimensional morphology, N-dimensional digital topology, N-dimensional thinning

1 Introduction

In [4-9], a general principle for morphological operations in cubic tessellated binary images X_N was described. One of its major applications is the skeletonization operation, which can be described as a conditional erosion operation. The conditions to the erosion are topology preservation conditions which are in this case constituted out of sets of structuring elements or mask. These sets can be partitioned in a number of sub-sets, e.g., for three dimensional images one can distinguish subsets that preserve volumes, surfaces, surface-ends, curves, curve-ends and single voxel objects. The subsets for topology preservation of surfaces and curves are made of so-called *shape primitives*, which are –in short- *all possible ways a non-bifurcating surface or a curve intersects a 3^3 neighbourhood*. We have elaborated on the connectivity paradox for high-dimensional images and, considering the fact that one likes to have the highest possible curvature to describe foreground objects, it is preferable to select the highest connectivity for foreground objects and the lowest for background objects. The topology preservation conditions are generated out of the shape primitives by properly intersecting foreground and background primitives. For three-dimensional images, the intersections of foreground surface primitives with background curve primitives, yield the surface preservation conditions; they specify all possible ways in which a foreground surface is possibly “pierced” by a background curve. This happens when

the central voxel swaps value from foreground to background. The same principle holds for the curve preservation conditions; they specify all possible ways in which a foreground curve is possibly “sliced” by a background surface. This, again, happens when the central voxel swaps value from foreground to background.

As we designed a general principle for skeletonization in cubic tessellated binary images X_N , based on the hit-or-miss transform [10], we came in [8] with a method that can be used to generate shape primitives up to X_N . It is a bootstrapping method, in which starting from a single voxel we gradually generate the shape primitives for ever higher object dimensions: A single voxel with two adjacent mutually not connected neighbour voxels form all possible space curve primitives. A voxel encircled by a closed space curve constitutes all surface primitives. We noticed also that as a consequence, the surface primitive is 18-connected and its behaviour can be described by a quadratic equation. This can also be used to measure the surface area of objects.

If we come to skeletonization in four dimensional images, the skeletonization procedure is identical to that in 3D and 2D, the question is only to find the suitable shape primitives. As was suggested in [8], the 3D procedure could be extended to 4D. In 4D we find space curves, curved surfaces, curved volumes and a hyper-volume. We stated that encircling a central voxel by surface patches generates the curved volumes; much alike a football is made out of leather patches. However, in practise this is a recursive procedure that takes almost infinite time to generate all possible situations. As a consequence, in this paper we describe a method to speed up the generation of shape primitives, starting from their quadratic descriptions.

2 Connectivities in N Dimensional Images

A set is connected if each pair of its points can be joined by a path along points that are all in the set, [3]. Within a square tessellated two-dimensional image with objects on a background, the objects can be chosen to consist of pixels connected with one or more of their 8 neighbours at (E, NE, N, NW, W, SW, S, SE) but as a consequence the background pixels must be connected with one or more of their 4 neighbours at (E, N, W, S) [1], or vice versa. This paradox extends to higher dimensions.

Tessellation in higher dimensions is also known as honeycombing. Coxeter [2] proved that the only regular honeycomb, i.e., a single regular identical cell on a lattice, which exists in all dimensions N, is the cubic honeycomb. This leaves hyper-cubic honeycombing as the only way to set-up a method that is able to operate from low to high dimensions. In [5], [6], we derived expressions to generate N^x , the connectivity between elements for N dimensional cases, i.e. the connectivities for 3^N neighbourhoods in 3D are N^6, N^{18}, N^{26} and in 4D are $N^8, N^{32}, N^{64}, N^{80}$. An object in a binary image X_N is defined as a set of mutually connected image elements with the same value. Generally the highest connectivity is chosen as foreground connectivity. Consequently, the connectivity of an *object* is the highest connectivity between any two adjacent elements (pixels, voxels, ...) within that object. If an image has more

than one object, we assume that all objects have the same value 1. The set of all objects in an image is the image foreground. When we define a basic object as a *non-forking* object with arbitrary shape and connectivity having a single intrinsic dimension \tilde{N} with: $0 \leq \tilde{N} \leq N$, we can categorize them as in Table 1.

Table 1. Intrinsic dimensions of basic objects

Dimension	Intrinsic dimension	Basic object type	Non-linear form
N=1	$\tilde{N} = 0$	Point	
	$\tilde{N} = 1$	Line	
N=2	$\tilde{N} = 0$	Point	
	$\tilde{N} = 1$	Line	Curve
	$\tilde{N} = 2$	Plane	
N=3	$\tilde{N} = 0$	Point	
	$\tilde{N} = 1$	Line	Space Curve
	$\tilde{N} = 2$	Planar	Curved Surface
	$\tilde{N} = 3$	Volume	
N=4	$\tilde{N} = 0$	Point	
	$\tilde{N} = 1$	Line	Space Curve
	$\tilde{N} = 2$	Planar	Curved Surface
	$\tilde{N} = 3$	Volume	Curved Volume
	$\tilde{N} = 4$	Hyper-Volume	

An example of a basic object is a non-bifurcating 18-connected curved surface in X_3 . Another example is a non-forking 26-connected space curve in X_3 . Note that curving is only possible when $1 \leq \tilde{N} \leq N - 1$. For $\tilde{N} = 0$ the basic object's size is the unit size, which prevents curving. For $\tilde{N} = N$ the basic object's structure spatially fills all dimensions and leaves no freedom for curving in another dimension. The degree of freedom for a basic object to curve is $N - \tilde{N}$. Basic objects are tessellated from shape primitives with a single intrinsic dimension \tilde{N} only.

In contrast with a basic object, we can define a compound object as an object of any arbitrary shape and size composed of one or more basic objects. Commonly a compound object is referred to as "object". It can bifurcate. Compound objects are tessellated from shape primitives from any intrinsic dimension. Examples in X_3 are objects composed of a number of volumes and / or surfaces and / or space curves, and in X_2 objects composed of a number of planes and / or curves.

Compound objects can be thinned from an intrinsic dimension \tilde{N} to a lower intrinsic dimension $\tilde{N} - 1$ by eroding them until the topology preservation conditions for dimensions $\tilde{N} \leq \tilde{N} - 1$ prevent further thinning. These conditions for intrinsic dimension \tilde{N} are found by intersecting foreground and background shape primitives of \tilde{N} .

As to each shape primitive, e.g., in 3D, an area or length contribution can be given, the total area of object boundaries or the length of space curves can be calculated by adding up the contributions of the primitive occurrences within an object.

3 The Generation of Shape Primitives in X_N

In [8], we defined Table 2, which gives per dimension N the relations between a shape primitive of object dimension \tilde{N} , the connectivity of the neighbour voxels to the Central Element (CE), the number of neighbour voxels connected to the central element (NE), and the recursive connectivity (RC). For example, a curved surface in 3D, $(N, \tilde{N}) = (3,2)$, has at least 4 neighbour voxels 18-connected to its central voxel. The neighbour voxels are mutually 26-connected (they are a closed space curve).

Table 2. Dimension, intrinsic dimension and connectivity for various shapes

N, \tilde{N}	Foreground shape primitive	CE	NE	RC	Background shape primitive	CE	NE	RC
(2,2)	Flat surface	4	4	-	Point	-	-	-
(2,1)	Curve	8	2	-	Curve	4	2	-
(3,3)	Flat volume	6	6	-	Point	-	-	-
(3,2)	Curved surface	18	≥ 4	26	Space curve	6	2	6
(3,1)	Space curve	26	2	-	Curved surface	18	≥ 4	6
(4,4)	Flat hyper-volume	8	8	-	Point	-	-	-
(4,3)	Curved volume	32	≥ 6	64	Space curve	80	2	8
(4,2)	Curved surface	64	≥ 4	80	Curved surface	64	≥ 4	8
(4,1)	Space curve	80	2	-	Curved volume	32	≥ 6	8

Table 2 specifies the algorithms with which the primitives can be generated:

- A curve primitive in X_3 is formed by a central voxel connected to two mutually not connected voxels. The curve roams over the 26-connected positions
- A curved surface primitive in X_3 is formed by a central voxel encircled by a closed space curve. This curve is 26-connected but is 18-connected to the central voxel
- A curve primitive in X_4 is formed by a central voxel connected to two mutually not connected voxels. The curve roams over the 80-connected positions.
- A curved surface primitive in X_4 is formed by a central voxel encircled by a closed space curve. The curve is 80-connected but it is 64-connected to the central voxel
- A curved volume primitive in X_4 is formed by a central voxel encircled by a closed set of surface patches. Each patch is a surface primitive. The surface patch is 64-connected but is 32-connected to the central voxel

These rules can be used to generate primitives as in figures 1-5. However, for dimensions higher than 3 this becomes cumbersome because of the recursive nature of the algorithms. Using quadratic equations to generate candidates for intrinsic dimensions $N = \tilde{N} - 1$ speeds up this generation considerably. Fig. 1 shows the set of space curve primitives for X_3 . Masks a and b form the 6-connected background set. Masks b..l form the 26-connected foreground set. Note that the masks {b, c, f, i} form the set of masks that is used as the foreground set of curves in X_2 , and the masks {a, b} form the background set in X_2 . The curves of the foreground set in X_2 are described (and can hence be generated) by:

$$y = ax + bx^2 \tag{1}$$

with $(a,b) \in \{0, \pm 0.5, \pm 1\}$ and $(|a+b| \leq 1)$

Which generates a set of 7 curves, of which 3 are mirrored versions of others:

<i>mask</i>	<i>(a,b)</i>	<i>equation</i>	<i>remark</i>
<i>b</i>	(0,0)	$y = 0$	
<i>c</i>	(1,0)	$y = x$	
<i>i</i>	(0,1)	$y = x^2$	
<i>f</i>	(0.5,0.5)	$y = 0.5x + 0.5x^2$	
-	(-1,0)	$y = -x$	<i>mirror of c</i>
-	(0,-1)	$y = -x^2$	<i>mirror of i</i>
-	(-0.5,-0.5)	$y = -0.5x - 0.5x^2$	<i>mirror of f</i>

(2)

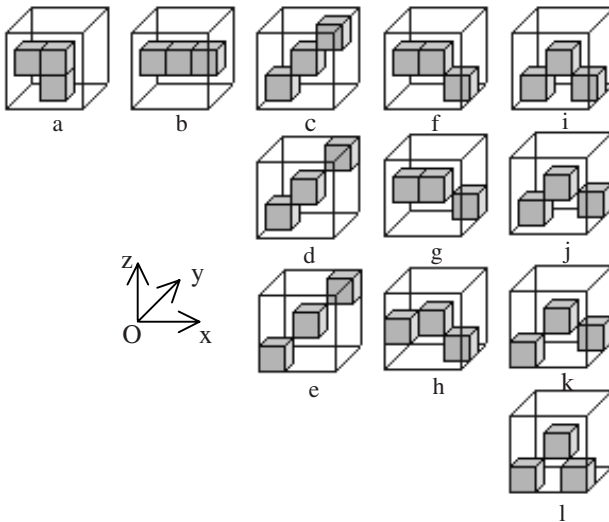


Fig. 1. The relation between the 4- and 8-connected curve primitives in X_2 and the 6- and 26-connected curve primitives in X_3

Observe in fig. 1 that the N^8 curve primitives in 2D {b, c, f, i} [4] are a subset of the N^{26} curve primitives in 3D. In X_2 they stay in the x-z plane. Note that another method to generate the 3D curve primitives {d, e, g, h, j, k, l} from the 2D curve primitives, is by exploring the newly obtained degree of freedom in the y direction. All non-central voxels are permuted over the ordinates $y = \{-1, 0, 1\}$. See, e.g., {j, k, l}, that have been derived from i.

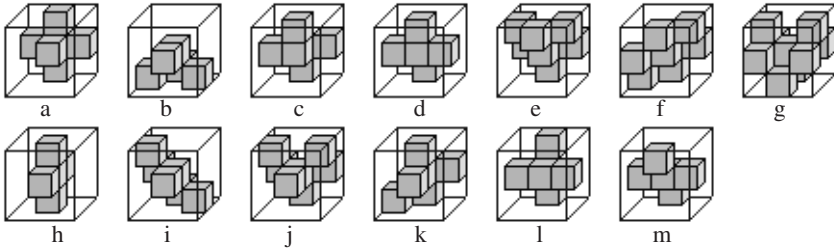


Fig. 2. The 18-connected surface primitives in X_3

Fig. 2 shows the set of foreground curved surface primitives for X_3 . In geometry, two lines span a plane. Likewise two space curves may span a curved surface. This is shown by the surface primitive of mask a, which is spanned by two perpendicular versions of mask i of fig. 1 (the quadratic mask). The reason for this is that the surfaces of the foreground set in X_3 are described (and can hence be generated) by:

$$z = ax + bx^2 + cy + dy^2 \tag{3}$$

with $(a, b, c, d) \in \{0, \pm 0.5, \pm 1\}$

and $(|a + b| \leq 1)$ and $(|c + d| \leq 1)$

$$\text{and } ((a \geq 0) \wedge (b \geq 0) \wedge (c \geq 0) \wedge (d \geq 0)) \vee ((b > 0) \wedge (c < 0)) \tag{4}$$

This generates a set of 22 surfaces, of which 9 are mirrored versions of others:

<i>mask</i>	<i>(a, b ; c, d)</i>	<i>equation</i>	<i>remark</i>
<i>h</i>	(0,0;0,0)	$z = 0$	
<i>m</i>	(1,0;0,0)	$z = x$	
<i>l</i>	(0,1;0,0)	$z = x^2$	
<i>d</i>	(0.5,0.5;0,0)	$z = 0.5x + 0.5x^2$	
<i>f</i>	(1,0;1,0)	$z = x + y$	
<i>e</i>	(1,0;0,1)	$z = x + y^2$	
<i>i</i>	(1,0;0.5,0.5)	$z = x + 0.5y + 0.5y^2$	

(5)

–	(0,1;1,0)	$z = x^2 + y$	<i>mirror of e</i>
a	(0,1;0,1)	$z = x^2 + y^2$	
b	(0,1;0.5,0.5)	$z = x^2 + 0.5y + 0.5y^2$	
–	(0.5,0.5;1,0)	$z = 0.5x + 0.5x^2 + y$	<i>mirror of i</i>
–	(0.5,0.5;0,1)	$z = 0.5x + 0.5x^2 + y^2$	<i>mirror of b</i>
c	(0.5,0.5;0.5,0.5)	$z = 0.5x + 0.5x^2 + 0.5y + 0.5y^2$	
–	(1,0;–1,0)	$z = x - y$	<i>mirror of f</i>
–	(1,0;0,–1)	$z = x - y^2$	<i>mirror of e</i>
–	(1,0;–0.5,–0.5)	$z = x - 0.5y - 0.5y^2$	<i>mirror of i</i>
–	(0,1;–1,0)	$z = x^2 - y$	<i>mirror of e</i>
g	(0,1;0,–1)	$z = x^2 - y^2$	
j	(0,1;–0.5,–0.5)	$z = x^2 - 0.5y - 0.5y^2$	
–	(0.5,0.5;–1,0)	$z = 0.5x + 0.5x^2 - y$	<i>mirror of i</i>
–	(0.5,0.5;0,–1)	$z = 0.5x + 0.5x^2 - y^2$	<i>mirror of b</i>
k	(0.5,0.5;–0.5,–0.5)	$z = 0.5x + 0.5x^2 - 0.5y - 0.5y^2$	

Fig. 3 shows some curved volume primitives for X_4 . Like in 3D, where two lines span a plane and two space curves may span a curved surface, in 4D two planes may span a volume and two curved surfaces may span a curved volume. The curved volumes of the foreground set in X_4 are described (and can hence be generated) by:

$$u = ax + bx^2 + cy + dy^2 + ez + fz^2 \tag{6}$$

with $(a,b,c,d,e,f) \in \{0, \pm 0.5, \pm 1\}$

and $(|a + b| \leq 1)$ and $(|c + d| \leq 1)$ and $(|e + f| \leq 1)$

$$\text{and } ((a \geq 0) \wedge (b \geq 0) \wedge (c \geq 0) \wedge (d \geq 0) \wedge (e \geq 0) \wedge (f \geq 0)) \vee \tag{7}$$

$$((b > 0) \wedge (d > 0) \wedge (f < 0))$$

Fig. 3 shows by way of example a flat volume a ($z = 0$), and the linear volumes¹ b ($z = y$), c ($z = u$), d ($z = x + y$), e ($z = y + u$) and f ($z = x + y + z$).

¹ Note that there is difference in terminology: in general, shape primitives of dimension $N = \tilde{N} - 1$ are indicated in the field of pattern recognition as hyper-planes, where we indicate them here as curved volumes. The term hyper-plane in pattern recognition is used to indicate that the structure separates the hyperspace in two parts, e.g. each of which contains a point cloud that represents a class. Our terminology indicates the basic shape of the objects

4 The Number of Unique Shape Primitives for $\tilde{N} = N - 1$

Quadratic equations of the form (8) can be used to generate the shape primitives of dimensions $\tilde{N} = N - 1$. It is of interest how many shape primitives can be found for each dimension.

$$X_{\tilde{N}} = \sum_{n=0}^{\tilde{N}} (a_n x_x + b_n x_n^2) \tag{8}$$

First, note that all linear solutions have two symmetry axes. Hence, for linear equations a difference version, e.g., $z = x - y$, is a mirrored version of the sum version, i.e., $z = x + y$. Secondly, all quadratic solutions have only one symmetry axis. But, a difference of a linear term and quadratic term will also produce mirrored versions. Only differences that involve two quadratic terms give rise to a new valid combination. This gives rise to the extra conditions (4) and (7) for the curved surfaces in 3D and the curved volumes in 4D.

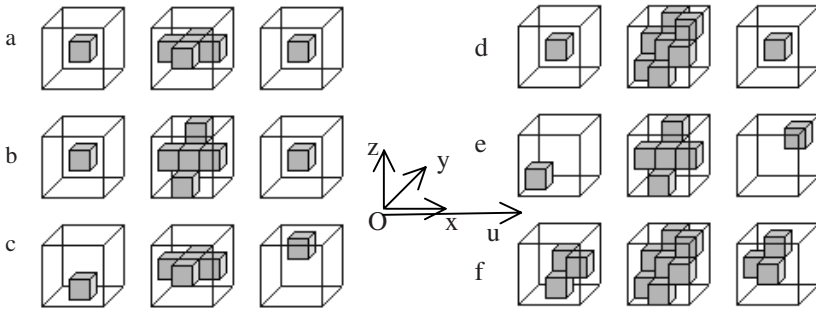


Fig. 3. Some 32 connected curved volume primitives in X_4

In order to estimate how many candidates we may expect for shape primitives of intrinsic dimension $\tilde{N} = N - 1$, we observe that in (8) the parameters come in tuples per dimension $(a,b),(c,d),\dots$, whereas each dimension adds a tuple. Each tuple has the four sum combinations $(0,0), (1,0), (0,1), (0.5,0.5)$, which we may code with $(0, 1, 2, 3)$. Consequently there are $4^{\tilde{N}}$ sum candidate combinations. In the case of the 3D surface primitives, this is 16. However, there are four self-mirrors $(ab, cd) = (00, 11, 22, 33)$, leaving 12 candidates that are asymmetric. Of those 12, $12/2 = 6$ are unique $(10 = 01, 20 = 02, 30 = 03, 21 = 12, 31 = 13, 32 = 23)$. Adding the four self-mirrors $(00, 11, 22, 33)$, yields as number of unique 3D surface primitives, based on a sum of terms, $C_{m+} = 10$. To set-up a general equation for the number of shape primitives, observe (12), in which the generating equation is setup for $\tilde{N} = 2$.

$$\begin{array}{cccc}
 i, j & i, j & i, j & i, j \\
 00 & & & \\
 10 & 11 & & \\
 20 & 21 & 22 & \\
 30 & 31 & 32 & 33
 \end{array} \Rightarrow C_{m+} = \sum_{i=0}^3 \sum_{j=1}^i (1) \tag{9}$$

This can be generalized for higher dimensions to:

$$C_{m+} = \sum_{n=0}^{\tilde{N}} \sum_{i=0}^n \sum_{j=0}^i (1) = \frac{1}{2} \sum_{n=0}^{\tilde{N}} (n^2 + 3n + 2) = \frac{1}{2} \sum_{n=0}^{\tilde{N}} (n+1)(n+2) \tag{10}$$

This sum can be solved to:

$$C_{m+} = \frac{1}{6} ((\tilde{N}+1)(\tilde{N}+2)(\tilde{N}+3)) \Rightarrow C_{m+} = \frac{N}{6} (N+2)(N+3) \tag{11}$$

This yields for intrinsic dimensions $N = \tilde{N} + 1 = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)$, the values of $C_{m+} = (1, 4, 10, 20, 35, 56, 84, 120, 165, 220, 286, 364)$. These numbers are also known as the Tetrahedral numbers from Pascal’s Triangle.

For the combinations involving differences of terms that all involve squares, each tuple has two combinations $(0, 1), (0.5, 0.5)$, which we may code with (a, b) . Consequently, there are in principle $2^{\tilde{N}}$ candidate combinations: (00, 01, 10, 11), in which the first and the last are self-mirrors and the middle two are mirrors of each other. This gives as number of unique 3D surface primitives based on a differences of terms: $C_{m-} = 3$. The general case is more complicated. For example, for $\tilde{N} = 2$, we have the sum $k+l$ and the difference $k-l$. For $\tilde{N} = 3$, we have the sum $k+l+m$ and the differences $k+l-m$ and $k-l-m$, which is, however, a mirrored version of the first difference. For $\tilde{N} = 4$, we have the sum $k+l+m+n$ and the differences $k+l+m-n$ and $k+l-m-n$ and $k-l-m-n$, where the first difference is a mirrored version of the last difference. If we write out the combinations of a and b for differences of terms for some intrinsic dimensions, we obtain Table 3. In this table bold a and b indicate a negative term, others a positive term. E.g., **aababb** indicates the term $-k-l-m+n+o+p$. Note that we have written down only half of the combinations. The lower half for each intrinsic dimension is identical to the upper half, i.e. for $\tilde{N} = 8$ the combinations for $(m, n) = (3, 5), (2, 6), (1, 7)$ and $(0, 8)$ must be added.

Note also the regular pattern in Table 3, which gives rise to the generating function of equation (12), but for odd \tilde{N} only. The second term in (12) is half of the total sum of the boxed entries in Table 3. The grey entries are omitted mirrored versions.

$$C_{m-} = \sum_{n=0}^{\tilde{N}} (\tilde{N}-1) + \left\lceil \frac{(m-1)(n-1)}{2} \right\rceil = \sum_{n=0}^{\tilde{N}} (\tilde{N}-1) + \left\lceil \frac{(\tilde{N}-n-1)(n-1)}{2} \right\rceil \tag{12}$$

Table 3. Combinations of $a^n b^m$ for intrinsic dimensions $\tilde{N} = 6, 7, 8$ with $m + n = \tilde{N}$

m	n	$\tilde{N} = 8$	
8	0	aaaaaaaa	4
7	1	aaaaaaab	7
6	2	aaaaaabb	7
		abaaaaab	3
5	3	aaaaabbb	7
		abaaaaab	2
		abbaaaab	2
4	4	aaaabbbb	7
		abaaabbb	2
		abbaaabb	2
		abbbaaab	1
			44
m	n	$\tilde{N} = 7$	
7	0	aaaaaaa	3
6	1	aaaaaab	6
5	2	aaaaabb	6
		abaaaaab	2
4	3	aaaabbb	6
		abaaabb	2
		abbaaab	1
			26
m	n	$\tilde{N} = 6$	
6	0	aaaaaa	3
5	1	aaaaaab	5
4	2	aaaabb	5
		abaaab	2
3	3	aaabbb	5
		abaabb	2
		abbaab	0
			22

This equation can be solved to (16):

$$C_{m-} = \frac{1}{12} \tilde{N}^3 + \frac{1}{2} \tilde{N}^2 - \frac{1}{12} \tilde{N} - \frac{1}{2} \quad \left\{ \text{while } C_{m+} \text{ was } \frac{1}{6} \tilde{N}^3 + \tilde{N}^2 + \frac{11}{6} \tilde{N} + 1 \right\} \quad (13)$$

For even dimensions N this gives:

$$C_{m,even} = \frac{1}{4} [N^3 + 3N^2 - 2N] \quad \Rightarrow \quad C_{m,even} = \frac{1}{4} [(N+1)(N+2) - 4] \quad (14)$$

Table 3 shows that for odd dimensions N , i.e. for even intrinsic dimensions \tilde{N} , when n is also even, there is a problem as the product of $(m-1)(n-1)$ is odd. For these dimensions we have to add a contribution $(\tilde{N}-1)/2$. This results in the equation:

$$C_{m,odd} = \frac{1}{4} [N^3 + 3N^2 - N + 1] \quad (15)$$

Written in one equation this gives:

$$C(N) = \frac{1}{4} \left[N^3 + 3N^2 - 2N + (N+1) \left(\left\lceil \frac{N}{2} \right\rceil - \left\lfloor \frac{N}{2} \right\rfloor \right) \right] \quad (16)$$

Consequently, the numbers for C_{m-} are: 0, 3, 6, 14, 22, 37, 52, 76, 100, 135, 170, while *the total number of unique $\tilde{N} = N - 1$ primitives for dimension $N = 1..12$* are:

$$C(1..12) = 1, 4, 13, 26, 49, 78, 121, 172, 241, 320, 421, 534 \quad (17)$$

5 The Unique Shape Primitives for $\tilde{N} < N - 1$

As fig. 1 showed, the curve primitives in 2D are a subset of the curve primitives in 3D. This procedure can also be used in higher dimensions to generate for example the ND curve primitives from the (N-1)D curve primitives by exploring the newly obtained degree of freedom.

Fig. 4 shows some masks from the set of foreground curved surface primitives for X_4 . The procedure to generate these masks is similar to the generation of space curves in X_3 using the space curves from X_2 . The surface masks from X_3 (figure 2), are placed in X_4 . The 3D surface primitives are a subset of the 4D surface primitives. They stay in X_4 in the x-y-z plane. The 4D masks can be generated from the 3D set by exploring the newly obtained degree of freedom in the u direction. All non-central voxels are permuted over the ordinates $u = \{-1, 0, 1\}$. Fig. 4 shows the permutation of mask 1 of fig. 2. In addition to this method, there is a downward method that is based on the observation that in 3D, the intersection of two planes is a line. Furthermore, the intersection of two curved surfaces may produce a space curve. This is not always the case, because two intersecting curved surfaces may partially overlap. As a consequence, this method can only be used to generate candidate situations for the shape primitives of $\tilde{N} < N - 1$. The candidates need to be sieved using the rules of section 3.

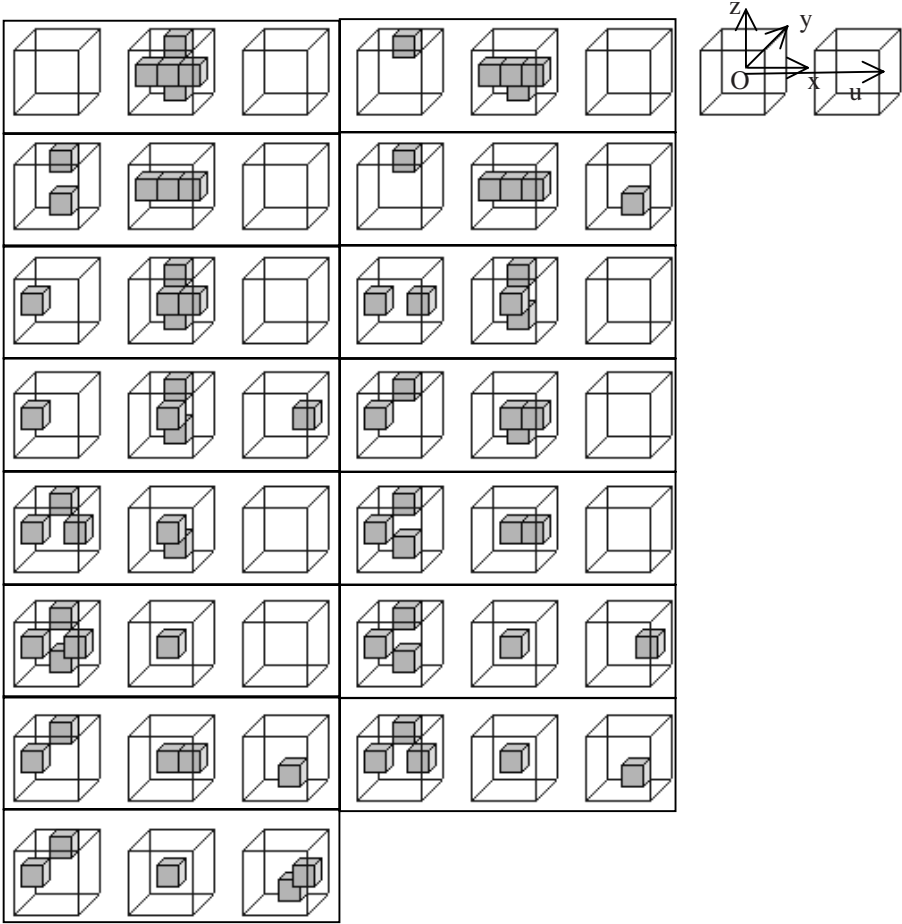


Fig. 4. Some 64-connected surface primitives in X_4

6 The Unique Shape Primitives for the Background

Fig. 5 shows the 6-connected surface primitives in X_3 , generated by taking a central voxel and encircling it with a 6-connected space curve that is 18-connected to the central voxel. This can be accelerated by using the quadratic equations of (18). However, as can be observed in fig. 5f, the z-axis runs along the body diagonal of the neighbourhood. Consequently, the points generated by (18) have to be rotated by rotating the body diagonal around the central voxel, $atan(\sqrt{2})$, in the direction of the z-axis. Fig. 6 shows for each mask (a.f) of fig 5. the result (a.f) of the surface generated by (18) and its rotated version.

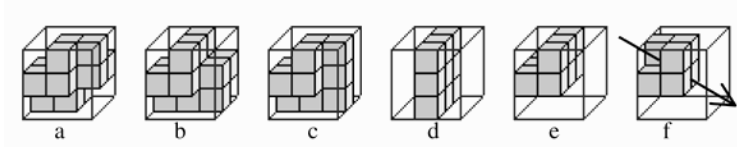


Fig. 5. The 6-connected surface primitives in X_3 , generated by encircling

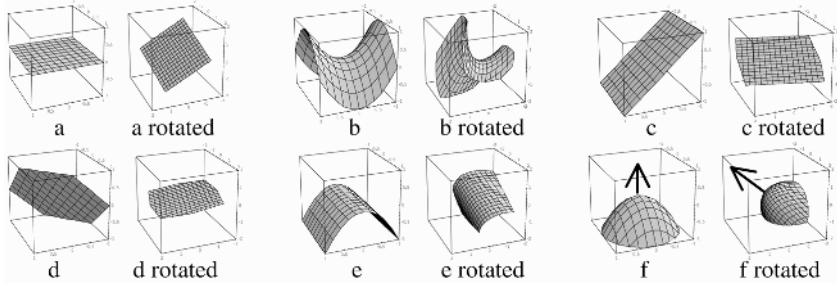


Fig. 6. The 6-connected surface primitives in X_3 , generated by (18) and rotation

<i>mask</i>	<i>(a,b;c,d)</i>	<i>equation</i>	<i>mask</i>	<i>(a,b;c,d)</i>	<i>equation</i>	(18)
<i>a</i>	(0,0;0,0)	$z = 0$	<i>d</i>	(1,0;1,0)	$z = x + y$	
<i>c</i>	(1,0;0,0)	$z = x$	<i>f</i>	(0,1;0,1)	$z = x^2 + y^2$	
<i>e</i>	(0,1;0,0)	$z = x^2$	<i>b</i>	(0,1;0,-1)	$z = x^2 - y^2$	

7 Conclusions

We described a method to accelerate the generation of shape primitives for N -dimensional images. These shape primitives can be used in conditions for topology preserving erosion or skeletonization in N -dimensional images, as well as for measurements. The method is based on the possibility to describe primitives for intrinsic dimensions $\tilde{N} = N - 1$ by quadratic equations. We derived an equation to predict the number of unique shape primitives thus generated, being $C(N) = (1, 4, 13, 26, 49, 78, 121)$ for the dimensions $N = \tilde{N} + 1 = 1..7$ Finally, we showed that derived from this method, also the primitive candidates for $\tilde{N} < N - 1$ as well as those for the background (lowest) connectivities can be found. This is used to speed up their generation.

Acknowledgement

This work was partially funded by the Swedish Foundation for International Cooperation in Research and Higher Education (STINT). The authors thank Jurjen Caarls for his fruitful contributions.

References

1. Rosenfeld A., Pfaltz J.L. (1966) Sequential operations in Digital Image Processing. *Journal of the ACM*, 471-494
2. Coxeter H.S.M. (1974) *Regular Polytopes* (3rd edition) Dover Publications, Inc. New York, ISBN 0-486-61480-8
3. Alexandroff P. (1961) *Elementary concepts of topology*. Dover Publications, Inc. New York, ISDN 0-486-60747-X
4. Jonker P.P., Komen E.R., Kraayveld M.A. (1995) A scalable, real-time image processing pipeline, *Machine Vision Applications*, (8): 110-121
5. Jonker P.P. (1992) *Morphological Image Processing: Architecture and VLSI design*. (Ph.D. Thesis) Kluwer Dordrecht, ISBN 90-2012766-7
6. P.P. Jonker and A.M. Vossepoel (1994), Connectivity in high dimensional images, *Proc. MVA'94, IAPR Workshop on Machine Vision Applications* (Kawasaki, Japan, Dec.13-15), IAPR, Kawasaki, Japan, 30-33
7. Jonker P.P., Vermeij O. (1996) On skeletonization in 4D images. In: P. Perner, P. Wang, A. Rosenfeld (eds.), *Advances in Structural and Syntactical Patt. Recogn.*, LNCS, vol. 1121, Springer Verlag, Berlin, 79-89
8. Jonker P.P. (2002), Skeletons in N dimensions using Shape Primitives, *Pattern Recognition Letters*, 23:677-686
9. Jonker P.P. (2003), Morphological Operations in Recursive Neighbourhoods, to appear in *Pattern Recognition Letters*
10. Serra J. (1982) *Image analysis and mathematical morphology*. Academic Press Inc., London
11. Soille P. (1999) *Morphological Image Analysis*, Springer Verlag, Berlin