Generating Similar Images Using Bag Context Shape Grammars

Blessing Ogbuokiri *Member, IAENG,* and Mpho Raborife

Abstract—**This paper presents four methods of generating an infinite number of similar images in a controlled manner using Bag Context Shape Grammars. Bag Context Shape Grammars are a class of shape grammars that uses context to control the application of a rule during derivation. These Bag Context Shape Grammar methods of image generation can be useful in generating similar images for application areas where similar images are needed, such as distractors for visual password scheme, decorative arts, CAPTCHA, and others.**

Index Terms—**Shape Grammar, Bag Context Grammar, Bag Context Shape Grammar, Image Generation, Image Generation Methods, Image Similarity.**

I. Introduction

A shape grammar is a class of context free grammars in formal languages that use spatial rules to generate images. Shape grammars have been studied for over two decades [1], [2], [3], [4], and have been applied mostly in architectural building plans, decorative art, engineering and product designs [5], [6].

Most shape grammar systems fail to generate only similar images in a controlled manner [7], [8], [9], [10]. For a shape grammar to generate similar images, it requires users to generate different images and manually select the similar images. It is difficult to know if the next image to be generated will be similar to a selected image [11], [12]. Existing researches that use shape grammars for the generation of images cannot regulate how similar images can be generated in a controlled manner [7], [8], [1], [2], [3], [9], [4], [10].

However, research has shown that control or regulation can be added to the context free grammar rules and thus to the derivation process using bag context [13], [12]. Bag context can be said to be a technique used to control when a context free grammar rule can be applied during a derivation [12]. This type of technique, when added to shape grammar rules, can also be used to produce or generate a gallery of images in a regulated manner. This technique is called Bag Context Shape Grammars (BCSGs) [14], [15].

BCSGs belong to the class of grammars that use context to regulate the application of context free rules [14]. In BCSGs, the application of a shape grammar (context free) rule is controlled by a special vector of integers called the bag, which changes during the derivation of an image [14]. A rule in the grammar can only be applied if the bag at that point is within the range defined by the lower and upper bounds of the rule. When a rule is applied, it causes the bag's values to change by adding the bag adjustment, which is part of the rule [14], [16], [15].

This paper, therefore, considers four methods of similar image generation using BCSGs. Images are considered similar when they are generated with the same grammar and they share some resemblance in appearance, such as colour or shape without being identical [15].

The four methods are as follows:

- Generation of images with variation: The variation here is the generation of images with a slight difference either in the structure or in the colour placement.
- Generation of images with sub–images.
- Generation of images with the same building levels but a different colour variation, and
- Generation of images with different building levels but a different colour variation.

The building level is the number of times shapes are added to generate an image or a sub–image from the beginning of the generation process. These methods of similar image generation are achievable with BCSGs.

The remainder of this paper is organized in the following relevant sections. Section II presents the BCSGs, followed by Section III that presents the four methods of similar image generation using BCSGs. Finally, Section IV is the conclusion and future work.

II. Bag Context Shape Grammars

The formal definition of BCSGs and example are adopted from [14], [17], and [16]. First, we present some notation and terms used in this section.

A. Notations and Terms

The symbol $\mathbb{N}$ represents the set of natural numbers {0, 1, 2, …}. $\mathbb{N}_+$ represents the set {1, 2, …}. For $k \in \mathbb{N}_+$, the set {1, 2, …, k} is denoted by $[k]$. $\mathbb{Z}$ represents the set of integers {…, -2, -1, 0, 1, 2, …}, then $\mathbb{Z}^2$ = $\{(x, y) \mid x, y \in \mathbb{Z}\}$. $\mathbb{Z}_\infty$ represents $\mathbb{Z} \cup$ {-∞, ∞}. If I = $[k]$, then elements of $\mathbb{Z}^I_\infty$ are written as $k-$tuples. If we have a vector of the form (k, k, …, k), then we denote it by **k**. For example, the vector (3, 3, …, 3) will be denoted by **3**.

A point, $(x,\ y)$ in $\mathbb{Z}^2$ is a position determined by a pair $(x,\ y)$, where $x, y \in \mathbb{Z}$. We denote a point by p. A line, l, is a finite length [18], [19]. In this part, a line segment, $\bar{l}$, is determined by any pair of two distinct points (p_1, p_2) on a line together with all the points of the line between p_1 and p_2, where p_1 and p_2 are called the endpoints [20], [21]. A shape, σ, is defined by the area enclosed by a finite set of line segments. A label can be denoted by any of the lowercase or uppercase letters, a–z and A–Z, respectively.

Manuscript received September 14th, 2020; revised July 11, 2021.

This research was supported by the Department of Science and Technology (DST) and the Council for Scientific and Industrial Research (CSIR) Inter–bursary support programme, South Africa.

Blessing Ogbuokiri holds a Ph.D.in Computer Science at the School of Computer Science and Applied Mathematics, University of the Witwatersrand, Johannesburg, South Africa. E-mail: ogbuokiriblessing@gmail.com.

Mpho Raborife is a Senior Lecturer at the Department of Applied Information Systems, University of Johannesburg, South Africa. E-mail: mraborife@uj.ac.za.

A labelled shape is a pair $(A,\ \sigma)$, where A is the label of the shape taken from the uppercase or lowercase letters and the symbol σ is as defined above. A grid is a coordinate plane consisting of a space of small squares, with a horizontal (x) axis and vertical (y) axis. Every labelled shape is presented in a unit square on the grid. The initial shape is usually labelled S. The labelled shape $(A,\ \sigma)$, is denoted by $(A,\ (x,\ y))$ in the rest of this work. Where $(x,\ y) \in \mathbb{Z}^2$ denotes the lower lefthand corner of the unit square the shape, σ, will be drawn.

A bag context shape grammar rule is commonly expressed in the form in Rule 1,

$$(A,\ (x,\ y)) \longrightarrow \{(A_1,\ (x_1,\ y_1)), (A_2,\ (x_2,\ y_2)), \ldots, (A_i,\ (x_i,\ y_i))\}\ (L_\mathrm{b}, U_\mathrm{b}; \delta) \qquad \text{(Rule 1)}$$

where A is a variable (uppercase label) and $(x,\ y) \in \mathbb{Z}^2$ is as defined above. The arrow "$\longrightarrow$" is interpreted as "transformed to", $A_1, A_2 \ldots A_i$ represent variable(s) and terminal(s) (lowercase labels), for $i \in \mathbb{N}_+$ and $(x_1,\ y_1)$, $(x_2,\ y_2)$, $\ldots$, $(x_i,\ y_i) \in \mathbb{Z}^2$. L_b, U_b represent the lower and upper bounds respectively, and δ is the bag adjustment.

The interpretation is as follows: a rule as in Rule 1 can be applied to a developing image if the developing image contains the variable A. Then, we take the set difference of the developing image and the variable A and take the set union of the developing image and $\{(A_1,\ (x_1,\ y_1)), (A_2,\ (x_2,\ y_2)), \ldots, (A_i,\ (x_i,\ y_i))\}$ provided the bag, β is within the range, $L_\mathrm{b} \leq \beta \leq U_\mathrm{b}$.

The number (n) of times a rule (r) is applied during the generative process is denoted by $r^{(n)}$. For instance, $2^{(5)}$ means that rule 2 is applied five times. The operations of shape union and difference treat shapes in the same basic way as the set-theoretic operations of union and difference treat sets. For visualisation purposes, every terminal is associated with a shape and the shape is filled with a chosen colour. The image is rendered to any size according to what is needed.

B. Formal Definition of Bag Context Shape Grammars

The definition of bag context shape grammars and example are taken from [14], [17], and [16].

Definition 1 *A BCSG, G = $(V_\mathrm{M},\ V_\mathrm{T},\ R,\ (S,\ (x,\ y)),\ I,\ \beta_0)$, has a finite alphabet V of labels, consisting of disjoint subsets V_M of variables and V_T of terminals. R is the set of rules, where every rule is of the form $(A, (x, y)) \longrightarrow \{(A_1, (x_1, y_1)), (A_2, (x_2, y_2)), \ldots, (A_i,\ (x_i,\ y_i))\}\ (L_\mathrm{b},\ U_\mathrm{b}\ ;\ \delta)$, where $A \in V_\mathrm{M}$, $\{A_1, A_2, \ldots, A_i\} \subseteq V$ and $(x,\ y)$, $(x_1,\ y_1)$, $\ldots$, $(x_i,\ y_i) \in \mathbb{Z}^2\ \forall\ i \in \mathbb{N}_+$, L_b, $U_\mathrm{b} \in \mathbb{Z}^I_\infty$, and δ is the bag adjustment, $\delta \in \mathbb{Z}^I$. $(S, (x,\ y))$ is the initial labelled shape, with $S \in V_\mathrm{M}$. I is the finite bag index set of the form $[k]$ and β_0 is the initial bag, $\beta_0 \in \mathbb{Z}^I$.*

Definition 2 *A pictorial form or evolving image is any set (composition) of labelled shapes in the plane denoted by Π. If Π is a pictorial form, we denote by $l(\Pi)$ the set of labels used in Π.*

Definition 3 *Let a configuration be a pair (Π, β) where Π is pictorial form and β is the bag. For a BCSG G and configurations (Π, β) and (Γ, β'), there is a derivation step from (Π, β) to (Γ, β'), if there is a rule $(s,\ (x,\ y)) \longrightarrow \{(s_1,\ (x_1,\ y_1)),\ (s_2,\ (x_2,\ y_2)),\ \ldots,\ (s_i,\ (x_i,\ y_i))\}\ (L_\mathrm{b},\ U_\mathrm{b}\ ;\ \delta)$ in R, where Π contains a labelled shape $(s,\ (x,\ y))$: $s \in V_M$ with $L_\mathrm{b} \leq \beta \leq U_\mathrm{b}$. $\Gamma = (\Pi \setminus \{(s,\ (x,\ y))\}) \cup \{(s_1, (x_1, y_1)), (s_2, (x_2, y_2)), \ldots, (s_i, (x_i, y_i))\}$ and $\beta' = \beta + \delta$.*

We denote the derivation step by $(\Pi,\ \beta) \Longrightarrow (\Gamma,\ \beta')$. This simply means that $(\Gamma,\ \beta')$ is directly derived from $(\Pi,\ \beta)$. If there is a sequence of zero or more derivation steps from $(\Pi,\ \beta)$ to $(\Gamma,\ \beta')$, then we denote that by $(\Pi,\ \beta) \Longrightarrow^* (\Gamma,\ \beta')$. We now say that $(\Gamma,\ \beta')$ is derived from $(\Pi,\ \beta)$.

Definition 4 *An image is a pictorial form Π with $l(\Pi) \subseteq V_T$ and the bag, β.*

Definition 5 *The gallery $\mathcal{G}(G)$ generated by a BCSG G is the set of images, Π, derivable from the initial labelled shape $(S,\ (x,\ y))$ and the initial bag β_0, represented as: $\mathcal{G}(G) = \{\Pi\colon ((S,\ (x,\ y)), \beta_0) \Longrightarrow^* (\Pi, \beta)$, for some $\beta \in \mathbb{Z}^I_\infty$ and $l(\Pi) \subseteq V_T\}$.*

Next, we demonstrate BCSG in Example II.1. Example II.1 generates images with the following: three legs of equal length and a spine. The second leg of the letter E to be generated divides the spine into an upper and a lower half which are the same length. See Fig. 1.

Example II.1 *Let $G_{\mathrm{E-bag}} = (V_M, V_T, R, (S,\ (x,\ y)), \{1, 2, 3, 4\}, 0)$, where $V_M = \{S,\ A,\ B,\ C,\ D,\ E\ \}$, $V_T = \{\ d\ \}$, $(S,\ (x,\ y)) = (S, (0, 0))$, and R is shown in Fig. 2.*

Some of the images in the gallery produced by $G_{\mathrm{E-bag}}$ when rendered using square shapes with the label d associated with dark colour are shown in Fig. 1. The images were scaled to the size as in Fig. 1 for presentation purposes.

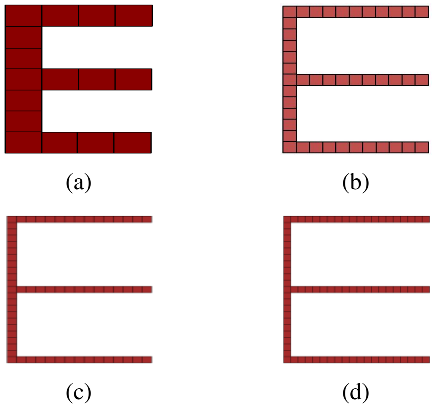

(a) (b) (c) (d)

Fig. 1: Some images in $\mathcal{G}(G_{\mathrm{E-bag}})$ [14], [17]

The strategy here is that the first, third, and fourth bag positions are used to control how the first, second and third legs grow. The second bag position is used to control the upper and lower spines of the letter E. The first (top) leg of the letter E is formed before any other part. The reason for this is that the number of times the rule that generates the top leg is applied is used to control the other legs. The upper spine is formed immediately after the top leg. The second leg is formed after the upper spine formation. Then the lower spine is formed followed by the third leg formation. The second leg must be exactly at half the spine which makes the upper and lower spines to be the same length. The three legs must also be the same length.

Next, we explain how the rules in Fig. 2 are applied to form the image in Fig. 1(a).

To achieve the strategy, the derivation starts with the initial labelled shape in the pictorial form, $\Pi = \{(S, (x, y))\}$. The application of Rule 2 of Fig. 2 replaces the initial labelled shape $(S, (x, y))$ with the shapes labelled $(d, (x, y))$, $(A, (x+1, y))$ and $(B, (x, y-1))$ in the set. The bag adjustment, (1, 1, 0, 0) is added to the bag, (0, 0, 0, 0) which is (0 + 1, 0 + 1, 0 + 0, 0 + 0) then the bag becomes (1, 1, 0, 0).

We can apply Rule 3 twice because the bag at each application is within the defined range. That is, the bag is greater than or equal to the lower bound, (1, 1, 0, 0) and less than or equal to the upper bound, ∞. The bag adjustment, (1, 0, 0, 0) is added to the bag at each application of Rule 3. This is followed by the application of Rule 4 because it is also within the defined range.

At this point, the bag is (2, 0, 0, 1). We apply Rule 5 twice to form the upper spine; the bag is (2, 2, 0, 1). Rule 6 is then applied to begin the formation of the second leg and the lower spine; the bag adjustment, (0, 0, 0, 1) is added to the bag, (2, 2, 0, 1) then the bag becomes (2, 2, 0, 2).

We further apply Rule 9 twice and Rule 10 once to form the second leg. At this point, the bag adjustment, (0, 0, 0, 1) is added to

$$R = \left\{ \begin{array}{lcll}
(S,(x,y)) & \longrightarrow & \{(d,(x,y)),(A,(x+1,y)),(B,(x,y-1))\}(0,0;(1,1,0,0)) & \text{(Rule 2)} \\
(A,(x,y)) & \longrightarrow & \{(d,(x,y)),(A,(x+1,y))\}((1,1,0,0),\infty;(1,0,0,0)) \mid & \text{(Rule 3)} \\
 & & \{(d,(x,y))\}((2,1,0,0),(\infty,\infty,\infty,1);(-1,-1,0,1)) & \text{(Rule 4)} \\
(B,(x,y)) & \longrightarrow & \{(d,(x,y)),(B,(x,y-1))\}((1,0,0,1),(\infty,\infty,\infty,1);(0,1,0,0)) \mid & \text{(Rule 5)} \\
 & & \{(d,(x,y)),(C,(x+1,y)),(D,(x,y-1))\} & \\
 & & \quad ((1,2,0,1),(\infty,\infty,0,1);(0,0,0,1)) & \text{(Rule 6)} \\
(D,(x,y)) & \longrightarrow & \{(d,(x,y)),(D,(x,y-1))\}((0,1,1,3),(0,\infty,\infty,3);(0,-1,0,0)) \mid & \text{(Rule 7)} \\
 & & \{(d,(x,y)),(E,(x+1,y))\}((0,0,1,3),(\infty,0,\infty,3);(0,0,0,1)) & \text{(Rule 8)} \\
(C,(x,y)) & \longrightarrow & \{(d,(x,y)),(C,(x+1,y))\}((1,1,0,2),(\infty,\infty,\infty,2);(-1,0,1,0)) \mid & \text{(Rule 9)} \\
 & & \{(d,(x,y))\}((0,1,2,2),(0,\infty,\infty,2);(0,0,0,1)) & \text{(Rule 10)} \\
(E,(x,y)) & \longrightarrow & \{(d,(x,y)),(E,(x+1,y))\}((0,0,1,4),(\infty,\infty,\infty,4);(1,0,-1,0)) \mid & \text{(Rule 11)} \\
 & & \{(d,(x,y))\}((1,0,0,4),(\infty,\infty,0,4);0) & \text{(Rule 12)}
\end{array} \right\}$$

Fig. 2: The set of rules for $G_{\text{E}-\text{bag}}$ in Example II.1. [14]

the bag, (0, 2, 2, 2) the bag becomes (0, 2, 2, 3). We apply Rule 7 twice followed by Rule 8 because the bag at each application is within the defined range. At this point, the bag adjustment, (0, 0, 0, 1), is added to the bag, (0, 0, 2, 3) then the bag becomes (0, 0, 2, 4) and the lower spine is formed.

Finally, we apply Rule 11 twice and Rule 12 once to form the third leg. The bag adjustment, (0, 0, 0, 0) is added to the bag, (2, 0, 0, 4), the bag becomes (2, 0, 0, 4). At this point, the picture formation is completed and the letter E is generated (see Fig. 1(a)).

We can also generate more images by repeating the process and applying Rules 3 and 5 as long as they are within the defined range at each application. This is because the number of times Rule 3 is applied determines the number of times Rules 9 and 11 will be applied and the number of times Rule 5 is applied determines the number of times Rule 7 will be applied with the help of the bag. Then, the process can be terminated with Rules 4, 10 and 12 respectively in order to form a complete picture (see Figs. 1(b)–(d)).

An illustration of one of the derivations of $G_{\text{E}-\text{bag}}$ using the picture in Fig. 1(a) is given in Fig. 3. The derivation has the lower lefthand corner of the initial shape S start at $x, y = 0$. The derivation is summarised in Table I showing the operations in the bag when an adjustment is made.

TABLE I: The bag during the derivation in Fig. 3.

Rules	The bag (β)	The bag adjustment (δ)
2	0	(1, 1, 0, 0)
$3^{(2)}$	(1, 1, 0, 0)	(1, 0, 0, 0)
4	(3, 1, 0, 0)	(-1, -1, 0, 1)
$5^{(2)}$	(2, 0, 0, 1)	(0, 1, 0, 0)
6	(2, 2, 0, 1)	(0, 0, 0, 1)
$9^{(2)}$	(2, 2, 0, 2)	(-1, 0, 1, 0)
10	(0, 2, 2, 2)	(0, 0, 0, 1)
$7^{(2)}$	(0, 2, 2, 3)	(0, -1, 0, 0)
8	(0, 0, 2, 3)	(0, 0, 0, 1)
$11^{(2)}$	(0, 0, 2, 4)	(1, 0, -1, 0)
12	(2, 0, 0, 4)	0

Next, we present the methods of similar image generation.

III. Methods of Similar Image Generation Using BCSGs

In this section, we demonstrate the four methods of similar image generation using BCSGs. These methods have become very necessary as the existing researches that use shape grammars for the generation of images cannot regulate how similar images can be generated in a controlled manner.

These methods are a generation of images with colour variation, generation of images with sub–images, generation of images with the same building levels but different colour variations, and generation of images with different building levels but a different colour variation.

A. Generation of Images with Colour Variation

Given an image in a BCSG generated gallery, there are many colours on the image. At all times during the derivation of the image, a colour must dominate a region of the image.

To achieve this method, consider a gallery that consists of images where each colour fills or dominates an entire row, see Example III.1.

Example III.1 *Let* $G_{\text{Carpet}-\text{A}} = (V_M, V_T, R, S, \{1, 2, 3\}, 0)$, *where* $V_M = \{S, A, B, C\}$, $V_T = \{d, w\}$, $(S,(x,y)) = (S,(0,0))$, *and* R *is the set in Fig. 5.*

We use the first and second bag positions to control the variables $A - C$ to ensure a colour dominates a row at a time. The third bag position ensures the number of rows is exactly the same as the number of columns. Some images generated using this example are shown in Fig. 4.

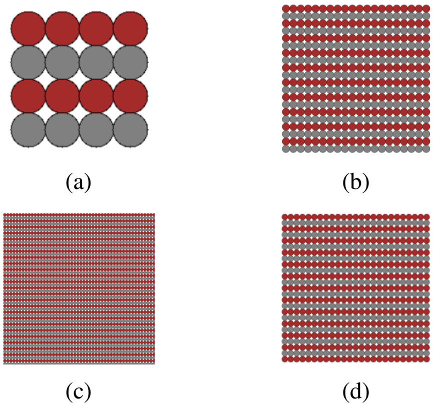

(a)
(b)
(c)
(d)

Fig. 4: Some of the images in $\mathcal{G}$ of $G_{\text{Carpet}-\text{A}}$ in Example III.1.

$$\begin{aligned}
\{(S,(0,0))\} &\overset{2}{\Rightarrow} \{(d,(0,0)),(A,(1,0)),(B,(0,-1))\}\\
&\xRightarrow{3^{(2)},4} \{(d,(0,0)),(d,(1,0)),(d,(2,0)),(d,(3,0)),(B,(0,-1))\}\\
&\xRightarrow{5^{(2)},6} \{(d,(0,0)),(d,(1,0)),(d,(2,0)),(d,(3,0)),(d,(0,-1)),(d,(0,-2)),\\
&\qquad (d,(0,-3)),(C,(1,-3)),(D,(0,-4))\}\\
&\xRightarrow{9^{(2)},10} \{(d,(0,0)),(d,(1,0)),(d,(2,0)),(d,(3,0)),(d,(0,-1)),(d,(0,-2)),\\
&\qquad (d,(0,-3)),(d,(1,-3)),(d,(2,-3)),(d,(3,-3)),(D,(0,-4))\}\\
&\xRightarrow{7^{(2)},8} \{(d,(0,0)),(d,(1,0)),(d,(2,0)),(d,(3,0)),(d,(0,-1)),(d,(0,-2)),\\
&\qquad (d,(0,-3)),(d,(1,-3)),(d,(2,-3)),(d,(3,-3)),(d,(0,-4)),\\
&\qquad (d,(0,-5)),(d,(0,-6)),(E,(1,-6))\}\\
&\xRightarrow{11^{(2)},12} \{(d,(0,0)),(d,(1,0)),(d,(2,0)),(d,(3,0)),(d,(0,-1)),(d,(0,-2)),\\
&\qquad (d,(0,-3)),(d,(1,-3)),(d,(2,-3)),(d,(3,-3)),(d,(0,-4)),\\
&\qquad (d,(0,-5)),(d,(0,-6)),(d,(1,-6)),(d,(2,-6)),\\
&\qquad (d,(3,-6))\}
\end{aligned}$$

Fig. 3: A derivation of the picture in Fig. 1(a) for $G_{\mathrm{E-bag}}$ [14].

$$R = \left\{ \begin{array}{lcll}
(S,(x,y)) & \longrightarrow & \{(d,(x,y)),(A,(x+1,y))\}(0,0;(1,0,1)) & \text{(Rule 13)}\\
(A,(x,y)) & \longrightarrow & \{(d,(x,y)),(A,(x+1,y))\}((1,0,1),(\infty,0,\infty);(1,0,1)) \mid & \text{(Rule 14)}\\
 & & \{(d,(x,y)),(B,(x,y-1))\}((2,0,2),(\infty,0,\infty);(0,0,-1)) & \text{(Rule 15)}\\
(B,(x,y)) & \longrightarrow & \{(B,(x-1,y)),(w,(x,y))\}((1,0,0),(\infty,\infty,\infty);(-1,1,0)) \mid & \text{(Rule 16)}\\
 & & \{(w,(x,y)),(C,(x,y-1))\}((0,1,1),(0,\infty,\infty);(0,0,-1)) \mid & \text{(Rule 17)}\\
 & & \{(w,(x,y))\}((0,(0,\infty,0);0) & \text{(Rule 18)}\\
(C,(x,y)) & \longrightarrow & \{(d,(x,y)),(C,(x+1,y))\}((0,1,1),(\infty,\infty,\infty);(1,-1,0)) \mid & \text{(Rule 19)}\\
 & & \{(d,(x,y)),(B,(x,y-1))\}((1,0,1),(\infty,\infty,\infty);(0,0,-1)) \mid & \text{(Rule 20)}\\
 & & \{(d,(x,y))\}((0,0,0),(\infty,0,0);0) & \text{(Rule 21)}
\end{array} \right\}$$

Fig. 5: The set of rules for $G_{\mathrm{Carpet-A}}$ in Example III.1.

The strategy in this example is that the number of times Rule 14 of Fig. 5 is applied determines the size of the image. Then, only one colour is allowed to dominate a row at a time.

To achieve the strategy, the application of Rule 13 of Fig. 5 starts the derivation. Then, Rule 14 is applicable as long as the bag at each application is within the defined range. That is, the bag is greater than or equal to the lower bound, (1, 0, 1) and less than or equal to the upper bound, (∞, 0, ∞). The bag adjustment, (1, 0, 1) is added to the bag which automatically increments the first and third bag positions by 1 at each application. This is to control the variables, while the number of rows and columns are also kept in check.

The application of Rule 15 adds the bag adjustment, (0, 0, -1) to the bag which automatically decrements the third bag position by 1. This also begins the formation of a new row. Then, Rules 16 and 17 followed by Rules 19 and 20 are applicable as long as the bag at each application is within the range. This process continues until the third bag position is 0, then, any of Rule 18 or 21 of Fig. 5 is applicable to complete the formation of the image. This strategy is sufficient to generate the images in Fig. 4. Next is the method of generating images with different parts.

B. Generation of Images with Sub-Images

Given a gallery of images generated by BCSG. There is another image in the upper right or upper left or lower right or lower left parts of the image called a sub–image.

In order to achieve this method, consider a gallery of images that consists of carpets with light or dark colour on the upper right and lower left parts. The top left part of each image in the gallery consists of dark and light colours. Each colour dominates a row. This feature is replicated on the lower right part of the image, see Fig. 6. We demonstrate this process in Example III.2.

Example III.2 *Let* $G_{\mathrm{Carpet-B}} = (V_M, V_T, R, S,$ $\{1, 2, 3, 4, 5\}, 0)$*, where* $V_M = \{S,\ A,\ B,\ C,\ A',\ A''\}$*,* $V_T = \{d,\ w\}$*,* $(S,(x,y)) = (S,(0,0))$*, and* R *is the set in Fig. 7. Some images in* $\mathcal{G}$ *produced by* $G_{\mathrm{Carpet-B}}$ *are shown in Fig. 6.*

The strategy here is that the variables A, A', and A'' in Example III.2 are used for the upper left part of the image. The variables B and C can be interchangeably used for the lower left and upper right parts respectively or the upper right part can also be mirrored on the lower left part depending on which rule is applied during derivation. Meanwhile, the lower right part of the image is a reflection of the upper left part. The derivation starts from the upper left part to the lower left part then to the upper right and

$$
\begin{array}{rcll}
R = \{ & & & \\
(S,(x,y)) & \longrightarrow & \{(d,(x,y)),(A,(x+1,y))\}(0,0;(1,0,1,0,0)) & \text{(Rule 22)} \\
(A,(x,y)) & \longrightarrow & \{(d,(x,y)),(A,(x+1,y))\}((1,0,1,0,0),(\infty,0,\infty,0,0);(1,0,1,0,0)) \mid & \text{(Rule 23)} \\
 & & \{(b,(x,y)),(B,(x+1,y),(A',(x-1,y))\} & \\
 & & \quad ((1,0,1,0,0),(\infty,0,\infty,\infty,0);(0,0,-1,1,0)) & \text{(Rule 24)} \\
 & & \{(b,(x,y)),(C,(x+1,y),(A',(x-1,y))\} & \\
 & & \quad ((1,0,1,0,0),(\infty,0,\infty,\infty,0);(0,0,-1,1,0)) & \text{(Rule 25)} \\
(A',(x,y)) & \longrightarrow & \{(A',(x-1,y)),(w,(x,y))\} & \\
 & & \quad ((1,0,0,0,0),(\infty,\infty,\infty,\infty,0);(-1,1,0,0,0)) \mid & \text{(Rule 26)} \\
 & & \{(w,(x,y)),(A'',(x,y-1))\} & \\
 & & \quad ((0,1,1,0,0),(0,\infty,\infty,\infty,0);(0,0,-1,1,0)) \mid & \text{(Rule 27)} \\
 & & \{(d,(x,y)),(B,(x,y-1))\}((1,0,0,0,0),(\infty,0,0,\infty,0);0) \mid & \text{(Rule 28)} \\
 & & \{(w,(x,y)),(C,(x,y-1))\}((0,1,0,0,0),(0,\infty,0,\infty,0);0) \mid & \text{(Rule 29)} \\
 & & \{(d,(x,y))\}(0,(0,\infty,0,\infty,\infty);0) & \text{(Rule 30)} \\
(A'',(x,y)) & \longrightarrow & \{(d,(x,y)),(A'',(x+1,y))\}((0,1,0,0,0),\infty;(1,-1,0,0,0)) \mid & \text{(Rule 31)} \\
 & & \{(d,(x,y)),(A',(x,y-1))\}((1,0,1,0,0),\infty;(0,0,-1,1,0)) \mid & \text{(Rule 32)} \\
 & & \{(d,(x,y)),(C,(x,y-1))\}((1,0,0,0,0),(\infty,0,0,\infty,0);0) \mid & \text{(Rule 33)} \\
 & & \{(d,(x,y)),(B,(x,y-1))\}((1,0,0,0,0),(\infty,0,0,\infty,0);0) \mid & \text{(Rule 34)} \\
 & & \{(d,(x,y))\}(0,(\infty,0,0,\infty,\infty);0) & \text{(Rule 35)} \\
(C,(x,y)) & \longrightarrow & \{(d,(x,y)),(C,(x+1,y))\} & \\
 & & \quad ((0,1,0,0,0),(\infty,\infty,\infty,\infty,0);(1,-1,0,0,0)) \mid & \text{(Rule 36)} \\
 & & \{(C,(x-1,y)),(d,(x,y))\} & \\
 & & \quad ((1,0,1,0,1),(\infty,\infty,\infty,\infty,1);(-1,1,0,0,0)) \mid & \text{(Rule 37)} \\
 & & \{(d,(x,y)),(C,(x,y-1))\} & \\
 & & \quad ((0,1,0,1,1),(0,\infty,\infty,\infty,1);(0,0,1,-1,-1)) \mid & \text{(Rule 38)} \\
 & & \{(d,(x,y)),(C,(x,y-1))\} & \\
 & & \quad ((1,0,0,1,0),(\infty,0,\infty,\infty,0);(0,0,1,-1,1)) \mid & \text{(Rule 39)} \\
 & & \{(w,(x,y)),(A'',(x,y-1))\} & \\
 & & \quad ((0,1,0,0,1),(0,\infty,0,\infty,1);(0,0,0,0,-1)) & \text{(Rule 40)} \\
 & & \{(w,(x,y)),(A',(x,y-1))\} & \\
 & & \quad ((1,0,0,0,0),(\infty,0,0,\infty,0);(0,0,0,0,0)) & \text{(Rule 41)} \\
 & & \{(d,(x,y))\}(0,(0,\infty,\infty,0,\infty);(0,0,0,0,-1)) & \text{(Rule 42)} \\
(B,(x,y)) & \longrightarrow & \{(w,(x,y)),(B,(x+1,y))\} & \\
 & & \quad ((0,1,0,0,0),(\infty,\infty,\infty,\infty,0);(1,-1,0,0,0)) \mid & \text{(Rule 43)} \\
 & & \{(B,(x+1,y),(w,(x,y)))\} & \\
 & & \quad ((1,0,1,0,1),(\infty,\infty,\infty,\infty,1);(-1,1,0,0,0)) \mid & \text{(Rule 44)} \\
 & & \{(w,(x,y)),(B,(x,y-1))\} & \\
 & & \quad ((1,0,0,0,0),(\infty,0,\infty,\infty,0);(0,0,-1,1,1)) \mid & \text{(Rule 45)} \\
 & & \{(w,(x,y)),(B,(x,y-1))\} & \\
 & & \quad ((0,1,1,0,1),(0,\infty,\infty,\infty,1);(0,0,-1,1,-1)) \mid & \text{(Rule 46)} \\
 & & \{(d,(x,y))\}(0,(0,\infty,\infty,0,\infty);(0,0,0,0,-1)) & \text{(Rule 47)} \\
 & & \{(w,(x,y)),(A'',(x,y-1))\}((0,1,0,0,1),(0,\infty,0,\infty,1);(0,0,0,0,-1)) & \text{(Rule 48)} \\
 & & \{(w,(x,y)),(A',(x,y-1))\}((1,0,0,0,0),(\infty,0,0,\infty,0);(0,0,0,0,0)) & \text{(Rule 49)} \\
\} & & &
\end{array}
$$

Fig. 7: The set of rules for $G_{\mathrm{Carpet-B}}$ in Example III.2.

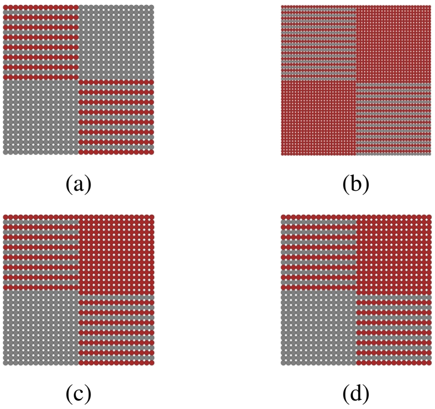


Fig. 6: Some of the images in $\mathcal{G}$ of $G_{\text{Carpet}-\text{B}}$ of Example III.2.

lower right parts of the image respectively. The size of the upper left parts of the image determines the size of the other three parts of the image. See Fig. 6.

To achieve the strategy, the first and second bag positions are used to control the light and dark colours respectively. The third bag position is used to ensure the number of rows is equal to the number of columns. The fourth and fifth bag positions are used to control the lower left, upper right and lower right part of the image respectively. The lower right part mimics the upper left part of the image. Such that, the number of light and dark colours in the upper left part is exactly the same in the lower right part. The size of the upper left part is duplicated on the lower left part with a dark or light colour. Then, the lower left part of the image is mirrored on the upper right part of the image with the same or an opposite colour depending on which rule is applied.

The application of Rule 23 of Fig. 7 after Rule 22 starts the formation of the upper left part of the image. The bag adjustment, (1, 0, 1, 0, 0) is added to the bag as long as the bag at each application is within the range, this continues until Rule 24 or 25 is applied. The application of Rule 24 or 25 begins the formation of a new row and creates room for the formation of the upper right part. Any of Rules 26–35 is applicable as long as the bag at each application is within the range. This process will form the upper left part of the image, then, begins the formation of the lower left part with the application of Rule 33 or 34. If Rule 33 is applied, then, Rules 36–42 are applicable when the bag is within the range at each application. On the other hand, if Rule 34 is applied, then, Rules 43–49 are applicable. The lower left part formation is completed by the application of Rule 42 or 47.

The formation of the upper right part begins with the application of Rule 36 or 43 of Fig. 7. The bag adjustment, (1, -1, 0, 0, 0) is added to the bag until the second bag position is 0. Then any of Rules 37–39 or 44–46 is applicable as long as the bag at each application is within the range. This process continues until the third bag position is 0, then, the upper right part is complete. The formation of the lower right part begins with the application of any of Rule 40, 42, 48 or 49, as long as the bag is within the range. At this point, any of Rules 26–35 is now applicable as long as the bag at each application is within the range as usual. This process continues until the lower right part formation is complete. Then the entire image formation is now complete. This strategy is sufficient to generate the images in Fig. 6. Next is the method of generating similar images by controlling the building levels and colour.

Example III.3 *We extend Example III.2 to generate a gallery of images that consists of light or dark colour on the top right and lower left corners. The top left corner of each image in the gallery consists of dark and light colours, such that each colour dominates a row. The lower right corner of the image consists of light and dark colours, such that no colour dominates a row. Hence the upper left and lower right corners (parts) of the image are not the same, see Figure 8.*

Let $G_{\text{Carpet}-\text{C}} = (V_M, V_T, R, S, \{1, 2, 3, 4, 5\}, 0)$, *where* $V_M = \{S, A, B, C, D, A', A'', D'\}$, $V_T = \{d, w\}$, $(S, (x, y)) = (S, (0, 0))$, *and R is the set in Figures 9(a) and (b) respectively.*

Some images in $\mathcal{G}$ produced by $G_{\text{Carpet}-\text{C}}$ are shown in Figure 8.

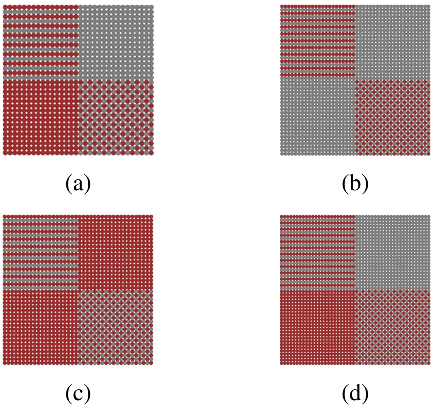


Fig. 8: Some of the images in $\mathcal{G}$ of $G_{\text{Carpet}-\text{C}}$ of Example III.3.

The strategy in Example III.3 follows the same process of the formation of the upper left, lower left, and upper right corners of the images in Example III.2. On the lower right corner of Example III.3 generated images, no colour is allowed to dominate a row or column, see Figure 8.

To achieve the strategy, the fifth bag position is used to control when to apply a dark or light colour. When the fifth bag position is 1 then a dark colour can be applied otherwise a light colour is applied. This ensures no colour is repeated immediately on a row or column after a rule that contains the colour has been applied. The lower right corner of the image is formed after all the other parts of the image have been completed. If the fifth bag position is 1 when the upper right corner of the image is formed then Rule 73 is applicable followed by Rules 82–85. If the fifth bag position is 0 when the upper right corner of the image is formed then Rule 75 is applicable followed by Rules 76–79.

This process continues until the third bag position is 0. The third bag position ensures the number of rows is equal to the number of columns. Then the process can be terminated by the application of any of the Rules 80, 81, 86, or 87 of Figure 9 as long as the bag is within the range at the time of application. This strategy is sufficient to generate the images in Figure 8.

C. Generation of Images with the Same Building Levels but Different Colour Variations

In a gallery of images generated by BCSG, there are many colour placements. A colour must dominate a part of the images. The number of times a shape is added to generate the images or part(s) of the images in the gallery must remain the same.

This can be achieved by limiting the number of some of the lower and upper bounds of the rules to the desired number. We Consider a grammar that some of the lower bounds or upper bounds are limited to 24 in order to enable the generation of images with a different colour place but the same building level, see Fig. 10. We demonstrate this method in Example III.4.

Example III.4 *Let a BCSG,* $G_{\text{SameBuild}} = (V_M, V_T, R, S, \{1, 2, 3, 4, 5\}, 0)$, *where* $V_M = \{S, A, B, C, D, A', A'', D', D''\}$, $V_T = \{d, w\}$, $(S, (x, y)) = (S, (0, 0))$, *and R is the set in Fig. 11. Some images in* $\mathcal{G}$ *produced by* $G_{\text{SameBuild}}$ *are shown in Fig. 10.*

.

The strategy in Example III.4 is that the first and second bag positions are used to control the colours. The third bag position

$$R = \Bigg\{$$

$$\begin{array}{rcll}
(S,(x,y)) & \longrightarrow & \{(d,(x,y)),(A,(x+1,y))\}(0,0;(1,0,1,0,0)) & \text{(Rule 50)} \\
(A,(x,y)) & \longrightarrow & \{(d,(x,y)),(A,(x+1,y))\}((1,0,1,0,0),(\infty,0,\infty,0,0);(1,0,1,0,0)) \mid & \text{(Rule 51)} \\
 & & \{(b,(x,y)),(B,(x+1,y),(A',(x-1,y))\} & \\
 & & \quad ((1,0,1,0,0),(\infty,0,\infty,\infty,0);(0,0,-1,1,0)) & \text{(Rule 52)} \\
 & & \{(b,(x,y)),(C,(x+1,y),(A',(x-1,y))\} & \\
 & & \quad ((1,0,1,0,0),(\infty,0,\infty,\infty,0);(0,0,-1,1,0)) & \text{(Rule 53)} \\
(A',(x,y)) & \longrightarrow & \{(A',(x-1,y)),(w,(x,y))\} & \\
 & & \quad ((1,0,0,0,0),(\infty,\infty,\infty,\infty,0);(-1,1,0,0,0)) \mid & \text{(Rule 54)} \\
 & & \{(w,(x,y)),(A'',(x,y-1))\} & \\
 & & \quad ((0,1,1,0,0),(0,\infty,\infty,\infty,0);(0,0,-1,1,0)) \mid & \text{(Rule 55)} \\
 & & \{(w,(x,y)),(B,(x,y-1))\}((0,1,0,0,0),(0,\infty,0,\infty,0);0) \mid & \text{(Rule 56)} \\
 & & \{(w,(x,y)),(C,(x,y-1))\}((0,1,0,0,0),(0,\infty,0,\infty,0);0) & \text{(Rule 57)} \\
(A'',(x,y)) & \longrightarrow & \{(d,(x,y)),(A'',(x+1,y))\}((0,1,0,0,0),\infty;(1,-1,0,0,0)) \mid & \text{(Rule 58)} \\
 & & \{(d,(x,y)),(A',(x,y-1))\}((1,0,1,0,0),\infty;(0,0,-1,1,0)) \mid & \text{(Rule 59)} \\
 & & \{(d,(x,y)),(B,(x,y-1))\}((1,0,0,0,0),(\infty,0,0,\infty,0);0) \mid & \text{(Rule 60)} \\
 & & \{(d,(x,y)),(C,(x,y-1))\}((1,0,0,0,0),(\infty,0,0,\infty,0);0) & \text{(Rule 61)} \\
(C,(x,y)) & \longrightarrow & \{(d,(x,y)),(C,(x+1,y))\} & \\
 & & \quad ((0,1,0,0,0),(\infty,\infty,\infty,\infty,0);(1,-1,0,0,0)) \mid & \text{(Rule 62)} \\
 & & \{(C,(x-1,y)),(d,(x,y))\} & \\
 & & \quad ((1,0,1,0,1),(\infty,\infty,\infty,\infty,1);(-1,1,0,0,0)) \mid & \text{(Rule 63)} \\
 & & \{(d,(x,y)),(C,(x,y-1))\} & \\
 & & \quad ((0,1,0,1,1),(0,\infty,\infty,\infty,1);(0,0,1,-1,-1)) \mid & \text{(Rule 64)} \\
 & & \{(d,(x,y)),(C,(x,y-1))\} & \\
 & & \quad ((1,0,0,1,0),(\infty,0,\infty,\infty,0);(0,0,1,-1,1)) \mid & \text{(Rule 65)} \\
 & & \{(d,(x,y))\}(0,(0,\infty,\infty,0,\infty);(0,0,0,0,-1)) \mid & \text{(Rule 66)} \\
 & & \{(w,(x,y)),(D',(x,y-1))\} & \\
 & & \quad ((0,1,0,0,1),(0,\infty,0,\infty,1);(0,0,0,0,-1)) \mid & \text{(Rule 67)} \\
 & & \{(w,(x,y)),(D,(x,y-1))\}((1,0,0,0,0),(\infty,0,0,\infty,0);(0,0,0,0,1)) & \text{(Rule 68)} \\
(B,(x,y)) & \longrightarrow & \{(w,(x,y)),(B,(x+1,y))\} & \\
 & & \quad ((0,1,0,0,0),(\infty,\infty,\infty,\infty,0);(1,-1,0,0,0)) \mid & \text{(Rule 69)} \\
 & & \{(B,(x+1,y),(w,(x,y)))\} & \\
 & & \quad ((1,0,1,0,1),(\infty,\infty,\infty,\infty,1);(-1,1,0,0,0)) \mid & \text{(Rule 70)} \\
 & & \{(w,(x,y)),(B,(x,y-1))\} & \\
 & & \quad ((1,0,0,0,0),(\infty,0,\infty,\infty,0);(0,0,-1,1,1)) \mid & \text{(Rule 71)} \\
 & & \{(w,(x,y)),(B,(x,y-1))\} & \\
 & & \quad ((0,1,1,0,1),(0,\infty,\infty,\infty,1);(0,0,-1,1,-1)) \mid & \text{(Rule 72)} \\
 & & \{(w,(x,y)),(D',(x,y-1))\} & \\
 & & \quad ((0,1,0,0,1),(0,\infty,0,\infty,1);(0,0,0,0,-1)) \mid & \text{(Rule 73)} \\
 & & \{(d,(x,y))\}(0,(0,\infty,\infty,0,\infty);(0,0,0,0,-1)) \mid & \text{(Rule 74)} \\
 & & \{(w,(x,y)),(D,(x,y-1))\}((1,0,0,0,0),(\infty,0,0,\infty,0);(0,0,0,0,1)) & \text{(Rule 75)}
\end{array}$$

$$\Bigg\}$$

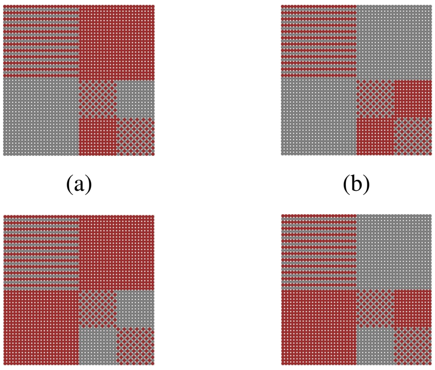
Fig. 9: (a) The set of rules for $G_{\mathrm{Carpet-C}}$.

$$R = \Bigg\{$$

$$\begin{array}{lcll}
(D,(x,y)) & \longrightarrow & \{(D,(x-1,y)),(d,(x,y))\} & \\
 & & \quad ((1,0,0,0,1),(\infty,\infty,\infty,\infty,1);(-1,1,0,0,-1))\ | & \text{(Rule 76)}\\
 & & \{(D,(x-1,y)),(w,(x,y))\} & \\
 & & \quad ((1,0,0,0,0),(\infty,\infty,\infty,\infty,0);(-1,1,0,0,1))\ | & \text{(Rule 77)}\\
 & & \{(b,(x,y)),(D',(x,y-1))\} & \\
 & & \quad ((0,1,0,1,1),(0,\infty,\infty,1,1);(0,0,1,-1,-1))\ | & \text{(Rule 78)}\\
 & & \{(w,(x,y)),(D',(x,y-1))\} & \\
 & & \quad ((0,1,0,0,0),(0,\infty,\infty,\infty,0);(0,0,1,-1,1))\ | & \text{(Rule 79)}\\
 & & \{(w,(x,y))\}(0,(\infty,\infty,\infty,0,0);0)\ | & \text{(Rule 80)}\\
 & & \{(d,(x,y))\}((0,0,0,0,1),(\infty,\infty,\infty,0,1);0) & \text{(Rule 81)}\\
(D',(x,y)) & \longrightarrow & \{(b,(x,y)),(D',(x+1,y))\} & \\
 & & \quad ((0,1,0,1,0),(\infty,\infty,\infty,0);(1,-1,0,0,1))\ | & \text{(Rule 82)}\\
 & & \{(w,(x,y)),(D',(x+1,y))\} & \\
 & & \quad ((0,1,1,0,1),(\infty,\infty,\infty,\infty,1);(1,-1,0,0,-1))\ | & \text{(Rule 83)}\\
 & & \{(w,(x,y)),(D,(x,y-1)),\} & \\
 & & \quad ((1,0,0,0,0),(\infty,0,\infty,\infty,0);(0,0,1,-1,1))\ | & \text{(Rule 84)}\\
 & & \{(d,(x,y)),(D,(x,y-1))\} & \\
 & & \quad ((1,0,1,1,1),(\infty,0,\infty,\infty,1);(0,0,1,-1,-1))\ | & \text{(Rule 85)}\\
 & & \{(w,(x,y))\}(0,(\infty,\infty,0,0,0);0)\ | & \text{(Rule 86)}\\
 & & \{(d,(x,y))\}((0,0,0,0,1),(\infty,\infty,\infty,0,1);0) & \text{(Rule 87)}
\end{array}$$

$$\Bigg\}$$

Fig. 9: (b) The set of rules for $G_{\text{Carpet}-\text{C}}$.



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

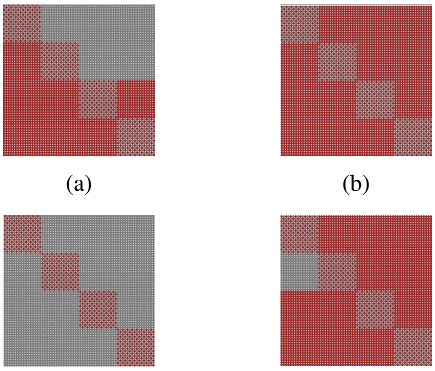
Fig. 10: Some of the images in $\mathcal{G}$ of $G_{\text{SameBuild}}$ in Example III.4.

ensures the number of rows and columns are the same. The fourth and fifth bag positions are used to control the parts of the image.

Observe that the highest value for some of the upper bounds in Fig. 11 is limited to 24, the same as the highest value for some of the lower bounds. Particularly, the first position of the lower bound in Rule 90 of Fig. 11. This helps to keep the images in Fig. 10 at the same building levels during derivation while the colour placement is different depending on the particular rule that is applied. Next, is the method of generating similar images with different building levels and colour variation.

More example of the above method in Section III-C is shown in Example III.5.

Example III.5 *Let a BCSG,* $G_{\text{SameBuild}-2} = (V_M, V_T, R, S, \{1, 2, 3, 4, 5\}, 0)$*, where* $V_M = \{S, A, B, C, D, A', A'', D', D''\}$*,* $V_T = \{d, w\}$*,* $(S,(x,y)) = (S,(0,0))$*, and* R *is equivalent to the set in Fig. 11 with some upper bound kept at 66. Some images in* $\mathcal{G}$ *produced by* $G_{\text{SameBuild}-2}$ *are shown in Fig. 12.*



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

Fig. 12: Some of the images in $\mathcal{G}$ of $G_{\text{SameBuild}-2}$ in Example III.5.

D. Generation of Images with Different Building Levels and Different Colour Variations

In this method, all images in a gallery generated by BCSG has a variation in the colour placements. A colour is allowed to dominate a part of the images. The number of times a shape is added to generate the images in the gallery should not be the same.

$$R = \{$$

$$\begin{array}{rcll}
(S,(x,y)) & \longrightarrow & \{(d,(x,y)),(A,(x+1,y))\}(0,0;(1,0,1,0,0)) & \text{(Rule 88)} \\
(A,(x,y)) & \longrightarrow & \{(d,(x,y)),(A,(x+1,y))\} & \\
 & & \quad((1,0,1,0,0),(24,0,24,0,0);(1,0,1,0,0)) \mid & \text{(Rule 89)} \\
 & & \{(b,(x,y)),(B,(x+1,y),(A',(x-1,y))\} & \\
 & & \quad((24,0,1,0,0),(24,0,24,24,0);(0,0,-1,1,0)) \mid & \text{(Rule 90)} \\
 & & \{(b,(x,y)),(C,(x+1,y),(A',(x-1,y))\} & \\
 & & \quad((24,0,1,0,0),(24,0,24,24,0);(0,0,-1,1,0)) & \text{(Rule 91)} \\
(A',(x,y)) & \longrightarrow & \{(A',(x-1,y)),(w,(x,y))\} & \\
 & & \quad((1,0,0,0,0),(24,24,24,24,0);(-1,1,0,0,0)) \mid & \text{(Rule 92)} \\
 & & \{(w,(x,y)),(A'',(x,y-1))\} & \\
 & & \quad((0,1,1,0,0),(0,24,24,24,0);(0,0,-1,1,0)) \mid & \text{(Rule 93)} \\
 & & \{(w,(x,y)),(B,(x,y-1))\}((0,1,0,0,0),(0,24,0,24,0);0) \mid & \text{(Rule 94)} \\
 & & \{(w,(x,y)),(C,(x,y-1))\}((0,1,0,0,0),(0,24,0,24,0);0) & \text{(Rule 95)} \\
(A'',(x,y)) & \longrightarrow & \{(d,(x,y)),(A'',(x+1,y))\}((0,1,0,0,0),24;(1,-1,0,0,0)) \mid & \text{(Rule 96)} \\
 & & \{(d,(x,y)),(A',(x,y-1))\}((1,0,1,0,0),24;(0,0,-1,1,0)) \mid & \text{(Rule 97)} \\
 & & \{(d,(x,y)),(B,(x,y-1))\}((1,0,0,0,0),(24,0,0,24,0);0) & \text{(Rule 98)} \\
 & & \{(d,(x,y)),(C,(x,y-1))\}((1,0,0,0,0),(24,0,0,24,0);0) & \text{(Rule 99)} \\
(C,(x,y)) & \longrightarrow & \{(d,(x,y)),(C,(x+1,y))\} & \\
 & & \quad((0,1,0,0,0),(24,24,24,24,0);(1,-1,0,0,0)) \mid & \text{(Rule 100)} \\
 & & \{(C,(x-1,y)),(d,(x,y))\} & \\
 & & \quad((1,0,1,0,1),(24,24,24,24,1);(-1,1,0,0,0)) \mid & \text{(Rule 101)} \\
 & & \{(d,(x,y)),(C,(x,y-1))\} & \\
 & & \quad((0,1,0,1,1),(0,24,24,24,1);(0,0,1,-1,-1)) \mid & \text{(Rule 102)} \\
 & & \{(d,(x,y)),(C,(x,y-1))\} & \\
 & & \quad((1,0,0,1,0),(24,0,24,24,0);(0,0,1,-1,1)) \mid & \text{(Rule 103)} \\
 & & \{(w,(x,y)),(D,(x,y-1))\} & \\
 & & \quad((0,1,0,0,0),(0,24,0,24,1);(0,0,0,0,-1)) \mid & \text{(Rule 104)} \\
 & & \{(d,(x,y))\}(0,(0,24,24,0,24);(0,0,0,0,-1)) & \text{(Rule 105)} \\
(B,(x,y)) & \longrightarrow & \{(w,(x,y)),(B,(x+1,y))\} & \\
 & & \quad((0,1,0,0,0),(24,24,24,24,0);(1,-1,0,0,0)) \mid & \text{(Rule 106)} \\
 & & \{(B,(x+1,y),(w,(x,y)))\} & \\
 & & \quad((1,0,1,0,1),(24,24,24,24,1);(-1,1,0,0,0)) \mid & \text{(Rule 107)} \\
 & & \{(w,(x,y)),(B,(x,y-1))\} & \\
 & & \quad((1,0,0,0,0),(24,0,24,24,0);(0,0,-1,1,1)) \mid & \text{(Rule 108)} \\
 & & \{(w,(x,y)),(B,(x,y-1))\} & \\
 & & \quad((0,1,1,0,1),(0,24,24,24,1);(0,0,-1,1,-1)) \mid & \text{(Rule 109)} \\
 & & \{(w,(x,y)),(D,(x,y-1))\} & \\
 & & \quad((0,1,0,0,0),(0,24,0,24,1);(0,0,0,0,-1)) \mid & \text{(Rule 110)} \\
 & & \{(d,(x,y))\}(0,(0,24,24,0,24);(0,0,0,0,-1)) & \text{(Rule 111)}
\end{array}$$

$$\}$$

Fig. 11: (a) The set of rules for $G_{\mathrm{SameBuild}}$ in Example III.4.

$R = \{$

$$
\begin{array}{rcll}
(D,(x,y)) & \longrightarrow & \{(d,(x,y)),(D,(x+1,y))\} & \\
 & & \quad ((0,1,0,1,0),(24,24,24,24,0);(1,-2,1,-2,1)) \mid & \text{(Rule 112)} \\
 & & \{(w,(x,y)),(D,(x+1,y))\} & \\
 & & \quad ((0,1,0,1,1),(24,24,24,24,1);(1,-2,1,-2,-1)) \mid & \text{(Rule 113)} \\
 & & \{(b,(x,y)),(D',(x,y-1)),(C,(x+1,y))\} & \\
 & & \quad ((1,0,1,0,0),(24,0,24,0,1);(0,0,-1,1,1)) \mid & \text{(Rule 114)} \\
 & & \{(w,(x,y)),(D',(x,y-1)),(B,(x+1,y))\} & \\
 & & \quad ((1,0,1,0,1),(24,0,24,0,1);(0,0,-1,1,-1)) & \text{(Rule 115)} \\
(D',(x,y)) & \longrightarrow & \{(D',(x-1,y)),(b,(x,y))\} & \\
 & & \quad ((1,0,0,1,0),(24,24,24,24,0);(-1,1,0,0,1)) \mid & \text{(Rule 116)} \\
 & & \{(D',(x-1,y)),(w,(x,y))\} & \\
 & & \quad ((1,0,0,1,1),(24,24,24,24,1);(-1,1,0,0,-1)) \mid & \text{(Rule 117)} \\
 & & \{(w,(x,y)),(D'',(x,y-1)),\} & \\
 & & \quad ((0,1,1,0,1),(0,24,24,24,1);(0,0,-1,1,-1)) \mid & \text{(Rule 118)} \\
 & & \{(d,(x,y)),(D'',(x,y-1))\} & \\
 & & \quad ((0,1,1,0,0),(0,24,24,24,0);(0,0,-1,1,1)) \mid & \text{(Rule 119)} \\
 & & \{(b,(x,y)),(C,(x,y-1))\} & \\
 & & \quad ((0,1,0,0,0),(0,24,0,24,0);0) \mid & \text{(Rule 120)} \\
 & & \{(w,(x,y)),(B,(x,y-1))\} & \\
 & & \quad ((0,1,0,0,1),(0,24,0,24,1);0) & \text{(Rule 121)} \\
D'',(x,y)) & \longrightarrow & \{(b,(x,y)),(D'',(x-1,y))\} & \\
 & & \quad ((0,1,0,1,0),(24,24,24,24,0);(1,-1,0,0,1)) \mid & \text{(Rule 122)} \\
 & & \{(w,(x,y)),(D'',(x-1,y))\} & \\
 & & \quad ((0,1,0,1,1),(24,24,24,24,1);(1,-1,0,0,-1)) \mid & \text{(Rule 123)} \\
 & & \{(b,(x,y)),(D',(x,y-1))\} & \\
 & & \quad ((1,0,1,0,0),(24,0,24,0,1);(0,0,-1,1,1)) \mid & \text{(Rule 124)} \\
 & & \{(w,(x,y)),(D',(x,y-1))\} & \\
 & & \quad ((1,0,1,0,1),(24,0,24,0,1);(0,0,-1,1,-1)) \mid & \text{(Rule 125)} \\
 & & \{(b,(x,y)),(B,(x,y-1))\} & \\
 & & \quad ((0,1,0,0,0),(0,24,0,24,0);0) \mid & \text{(Rule 126)} \\
 & & \{(w,(x,y)),(C,(x,y-1))\} & \\
 & & \quad ((0,1,0,0,1),(0,24,0,24,1);0) & \text{(Rule 127)}
\end{array}
$$

$\}$

Fig. 11: (b) The set of rules for $G_{\text{SameBuild}}$ in Example III.4.

To achieve this method, consider a gallery of images with a variation in colour, such that the number of the light colour on the upper part of the image is half the number of the light colour on the lower part of the images. The difference between the images is the variations in the building levels and colour placement, see Fig. 13. We demonstrate this method in Example III.6.

Example III.6 *Let* $G_{\text{DiffBuild}} = (V_M, V_T, R, S,$ $\{$*1, 2, 3, 4, 5*$\}, 0)$*, where* $V_M = \{$*S, A, B, C*$\}$*,* $V_T = \{$*d, w*$\}$*,* $(S,(x,y)) = (S,(0,0))$*, and* R *is the set in Fig. 14.*

The strategy in Example III.6 is that the first and second bag positions are used to control the variables A–C which help to keep colour on a row. The third bag position is used to ensure the number of rows is the same as the number of columns. The fourth and fifth bag positions are used to ensure the number of the light colour on the lower left part of the last dark row of the image is twice the number of the light colour on the upper right part of the first dark row of the image. This is achieved by limiting some of the lower bounds to 30 and some of the upper bounds in the rules to 35. Particularly, Rules 130, 131, and 133 of Fig. 14.

This means that Rules 131 and 133 of Fig. 14 can only be applied when the values in the bag positions one and three are 30 respectively. This will help to generate only images that are similar to the images in Fig. 13.

Some images in $\mathcal{G}$ produced by $G_{\text{DiffBuild}}$ in Example III.6 are shown in Fig. 13.

We repeat the approach used to generate the images in Figure 13 in Example III.7.

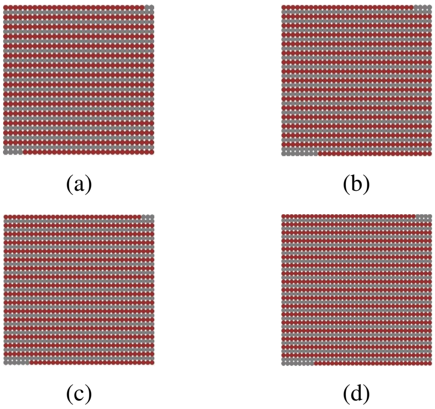
Example III.7 *Let* $G_{\text{DiffBuild}-2} = (V_M, V_T, R, S,$ $\{$*1, 2, 3, 4, 5*$\}, 0)$*, where* $V_M = \{$*S, A, B, C*$\}$*,* $V_T = \{$*d, w*$\}$*,* $(S,(x,y)) = (S,(0,0))$*, and* R *is equivalent to the set in Fig. 14 with some of the upper bound set to 20, 30, 80, and 90 respectively.*

$$R = \Bigg\{$$

$$\begin{array}{rcll}
(S,(x,y)) & \longrightarrow & \{(d,(x,y)),(A,(x+1,y))\}(0,0;(1,0,1,0,0)) & \text{(Rule 128)} \\
(A,(x,y)) & \longrightarrow & \{(d,(x,y)),(A,(x+1,y))\} & \text{(Rule 129)} \\
 & & \quad ((1,0,1,0,0),(35,0,35,0,0);(1,0,1,0,0)) \mid & \text{(Rule 130)} \\
 & & \{(w,(x,y)),(B,(x,y-1))\} & \\
 & & \quad ((30,0,30,0,0),(35,0,35,35,35);(0,0,-1,1,0)) \mid & \text{(Rule 131)} \\
 & & \{(w,(x,y)),(A,(x+1,y))\} & \text{(Rule 132)} \\
 & & \quad ((30,0,30,0,0),(35,0,35,35,35);(1,0,1,1,0)) & \text{(Rule 133)} \\
(B,(x,y)) & \longrightarrow & \{(B,(x-1,y)),(w,(x,y))\} & \text{(Rule 134)} \\
 & & \quad ((1,0,0,0,0),(35,35,35,35,0);(-1,1,0,0,0)) \mid & \text{(Rule 135)} \\
 & & \{(w,(x,y)),(C,(x,y-1))\} & \text{(Rule 136)} \\
 & & \quad ((0,1,1,1,0),(0,35,35,35,0);(0,0,-1,0,0)) & \text{(Rule 137)} \\
(C,(x,y)) & \longrightarrow & \{(d,(x,y)),(C,(x+1,y))\} & \text{(Rule 138)} \\
 & & \quad ((0,1,1,1,0),((35,35,35,35,0));(1,-1,0,0,0)) \mid & \text{(Rule 139)} \\
 & & \{(d,(x,y)),(B,(x,y-1))\} & \text{(Rule 140)} \\
 & & \quad ((1,0,1,1,0),((35,0,35,35,0));(0,0,-1,0,0)) \mid & \text{(Rule 141)} \\
 & & \{(w,(x,y)),(C,(x+1,y))\} & \text{(Rule 142)} \\
 & & \quad ((0,1,0,1,0),(35,35,0,35,35);(0,-1,0,-1,1)) \mid & \text{(Rule 143)} \\
 & & \{(w,(x,y)),(C,(x+1,y))\} & \text{(Rule 144)} \\
 & & \quad ((0,1,0,0,1),(35,35,0,0,35);(0,-1,0,0,-1)) \mid & \text{(Rule 145)} \\
 & & \{(d,(x,y)),(C,(x+1,y))\} & \text{(Rule 146)} \\
 & & \quad ((0,1,0,0,0),(35,35,0,0,0);(0,-1,0,0,0)) \mid & \text{(Rule 147)} \\
 & & \{(d,(x,y))\}(0,0;0) & \text{(Rule 148)}
\end{array}$$

$$\Bigg\}$$

Fig. 14: The set of rules for $G_{\text{DiffBuild}}$ in Example III.6.



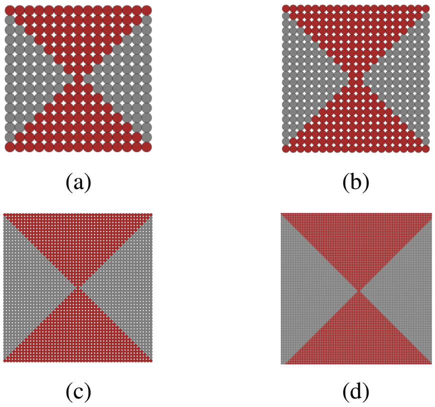
(a) (b) (c) (d)

Fig. 13: Some of the images in $\mathcal{G}$ of $G_{\text{DiffBuild}}$ in Example III.6.

Some images in $\mathcal{G}$ produced by $G_{\text{DiffBuild}-2}$ in Example III.7 are shown in Fig. 15.

Next, we adjust the set in Fig. 14 to generate the type of images in Fig.16, see Example III.8.

Example III.8 *Let $G_{\text{DiffBuild}-3} = (V_M, V_T, R, S,$ $\{$1, 2, 3, 4, 5$\}$, 0), where $V_M = \{S,\ A,\ B,\ C\}$, $V_T = \{d,$ $w\}$, $(S,(x,y)) = (S,(0,0))$, and R is equivalent to the set in Fig. 14 with some of the upper bound set to 40, 60, 70, and 80 respectively.*



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

Fig. 15: Some of the images in $\mathcal{G}$ of $G_{\text{DiffBuild}-2}$ in Example III.7.

Some images in $\mathcal{G}$ produced by $G_{\text{DiffBuild}-3}$ in Example III.8 are shown in Fig. 16.

IV. Conclusion and Future Work

In this paper, BCSGs was presented with an example to show how they can generate images. Further, the four BCSG methods of generating similar images were demonstrated. These methods include the generation of images with colour variation and the generation of images with different parts. Others are, generation of similar images with the same building level but different colour

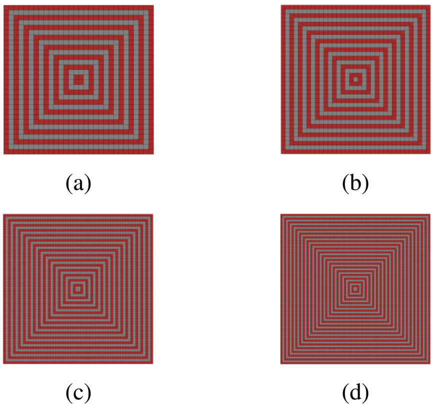


Fig. 16: Some of the images in $\mathcal{G}$ of $G_{\text{DiffBuild}-3}$ in Example III.8.

placement and generation of similar images with different building levels but different colour placement, these were achieved by limiting some of the lower and upper bounds of a given set of rules.

It was observed that existing shape grammars lacked the capacity to control the generation of similar images in these manners. With the above methods of similar image generation using BCSGs, it is possible to know the next similar image to be generated to a selected image, which other shape grammars lack the ability to do.

In the future, we intend to demonstrate the power of these methods by showing how they can be used to generate similar images in application areas where the use of similar images is needed, such as the generation of distractor images for visual password scheme, decorative art, CAPTCHA, etc.

REFERENCES

[1] G. Stiny, “Introduction to shape and shape grammars,” *Environment and Planning B*, vol. 7, no. 3, pp. 343–351, 1980.

[2] ——, “What is a design?” *Environment and Planning B*, vol. 17, pp. 97–103, 1990.

[3] ——, “Weights,” *Environment and Planning B*, vol. 19, pp. 413–430, 1992.

[4] G. Stiny and W. J. Mitchell, “The Palladian grammar,” *Environment and Planning B: Planning and Design*, vol. 5, no. 1, pp. 5–18, 1978.

[5] S. Ahmad and S. Chase, “Transforming grammars for goal driven style innovation,” in *Predicting the future, Proceedings of the 25th Conference on Education in Computer Aided Architectural Design in Europe (eCAADe), Frankfurt am Main, Germany*, 2007, pp. 879–886.

[6] S. Arida, “Volkswagen Beetle Grammar. Computational design, Final MSc project, Massachusetts Institute of Technology (MIT),” 2002.

[7] T. W. Knight, “Color grammars: the representation of form and color in design,” *Leonardo*, vol. 26, pp. 117–124, 1993.

[8] ——, “Shape grammar and color grammar in design,” *Environment and Planning B: Urban Analytics and City Science*, vol. 21, no. 6, pp. 705–735, 1994.

[9] G. Stiny, “How to calculate with shapes,” *Formal engineering design synthesis*, vol. 19, pp. 13–27, 2001.

[10] R. Stouffs, “Description grammars : A general notation,” *Environment and Planning B: Urban Analytics and City*, vol. 45, no. 1, pp. 106–123, 2016.

[11] N. Jingili, S. Ewert, and I. Sanders, “Measuring perceptual similarity of syntactically generated pictures,” in *8th International Conference on Simulation and Modelling Methodologies, Technology and Application (SIMULTECH), 30 July, Porto, Portugal*, 2018, accessible http://insticc.org/node/TechnicalProgram/simultech/ presentationDetails/69065, [Last accessed 20-March-2019].

[12] S. Ewert, N. Jingili, L. Mpota, and I. Sanders, “Bag context picture grammars,” *Journal of Computer Languages*, 2019, doi: https://doi.org/10.1016/j.cola.2019.04.001.

[13] F. Drewes, C. du Toit, S. Ewert, B. van der Merwe, and A. P. van der Walt, “Bag context tree grammars,” *Developments in Language Theory*, vol. 4036, pp. 226–237, 2006.

[14] B. Ogbuokiri and M. Raborife, “Bag context shape grammars,” *IAENG Journal of Computer Science*, vol. 47, no. 1, pp. 75–86, 2020.

[15] ——, “The similarity of images generated by bag context shape grammars,” in *Proceedings of the 2020 International SAUPEC/RobMech/PRASA Conference*. Cape Town, 29–31 January, 2020, pp. 1–6.

[16] ——, “Bag context shape grammar implementation: From theory to useable software,” *Computer-Aided Design and Applications*, vol. 17, no. 3, pp. 548–574, 2019.

[17] B. Ogbuokiri, “Generation of similar images using bag context shape grammars,” PhD Thesis submitted to the School of Computer Science and Applied Mathematics, University of the Witwatersrand, Johannesburg. SA, 2020.

[18] C. Anthony and D. Robert, *Foundation Maths*, 6th ed. Edinburgh Gate, United Kingdom: Pearson Education Limited, 2016.

[19] C. Khalil and B. A. Jacques, “The maximum (k,m)-subsets problem is in the class nexp,” *IAENG International Journal of Computer Science*, vol. 48, no. 2, pp. 451–455, 2021.

[20] Z. Tiefeng, “Application of a new model for complex systems,” *Engineering Letters*, vol. 29, no. 2, pp. 471–476, 2021.

[21] H. M. Wali, “Compactness of hausdorff fuzzy metric spaces,” *IAENG International Journal of Applied Mathematics*, vol. 51, no. 1, pp. 17–24, 2021.