A Topological Approach to Creating Any *Pulli Kolam*, an Artform from South India

Venkatraman Gopalan* and Brian K. Vanleeuwen

Materials Science and Engineering, Pennsylvania State University, University Park, PA 16802, U.S.A.
**E-mail address: vgopalan@psu.edu*

(Received March 7, 2015; Accepted August 20, 2015)

Pulli kolam is a ubiquitous art form drawn afresh every morning on the threshold of most homes in South India. It involves drawing a line looped around each dot of a collection of dots (*pulli*s) placed on a plane in accordance with three mandatory rules, namely, all dots should be circumscribed, all line orbits should be closed, and two line segments cannot overlap over a finite length. The mathematical foundation for this art form has attracted attention over the years. In this work, we propose a simple 5-step method by which one can systematically draw all possible *kolam*s for any number of dots N arranged in any spatial configuration on a surface. For a given N, there exist a set of parent *kolam*s from which all other possible *kolam*s can be derived. All parent *kolam*s arising from different spatial arrangements of N dots can be classified into parent *kolam* types; within each type, all parents are topologically equivalent, or homotopic. The number of *kolam*s for a given N is shown to be infinite if only the three mandatory rules stated above are followed; it becomes finite as more optional rules and restrictions are imposed. This intuitive method can be mastered by anyone to create countless *kolam*s with no prior knowledge or the need for a detailed mathematical understanding. It is also amenable to developing apps and educational games that introduce the concepts of symmetry and topology.

Key words: *Kolam*, Art, South India, Topology, Homotopy

1. What is a *Kolam*?

Figure 1 depicts an example of a *kolam*, an ancient and still popular South Indian art form. This particular type of *kolam* is called the *pulli kolam* in Tamil, which consists of a series of dots (called *pulli*s) placed on a surface, each of which is then circumscribed by lines that form closed orbits. It is a very common sight on the threshold of homes in the five southern states with a combined current population of ∼252 million. They are called by varied names in the respective regional languages of these states: *kolam* in Tamil spoken in Tamil Nadu, *golam* in Malayalam spoken in Kerela, *rangole* in Kannada spoken in Karnataka, and *muggulu* in Telugu spoken in Andhra Pradesh and Telangana. With every sunrise, women wash the floor in front of the houses, and using rice flour, place the dots and draw a *kolam* largely from memory. Learning how to draw *kolam*s from an early age is an important aspect of growing up in southern India, especially for girls. As they continue to learn from other women in their family, the *kolam*s become increasingly complex, with a larger number of dots and more intricate line orbits. Remembering the dot configurations and line orbits is a daily exercise in geometric thinking. The process is immensely pleasurable, especially when a *kolam* is successfully completed with no loose ends.

While the conventional *kolam*s impose several rules, here we begin with three simple rules in order to give ourselves greater room for discovery and creativity. Given an arbitrary arrangement of dots on a plane, the following three *mandatory* (**M**) rules define a *kolam*:

M1: All dots should be circumscribed.
M2: All interactions between two lines must be at points, i.e. two line segments cannot overlap over a finite length.
M3: All line orbits should be closed, i.e. no loose ends.

In addition to the above rules, one may choose to apply additional *optional* (**O**) rules. There is no limit to the number of such optional rules that can be followed, but we will explore some of them later in this work.

While *kolam*s are widely rendered from memory, the process of creating entirely new ones, especially complex *kolam*s with a large number of dots is far more challenging. This work attempts to provide a simple 5-step method by which anyone can create a very large number of *kolam*s from any arbitrary pattern of dots. The proposed topological method deemphasizes memory; in principle, anyone who knows just the method will be able to draw a large number of *kolam*s with no other prior knowledge.

Many previous pioneering works exist that have provided mathematical insights into the form of a *kolam* over the past four decades. These include converting *kolam*s into numbers and linear diagrams [1], using graph, picture, and array grammers [2–10], extended pasting schemes [11], morphism of monoids [12], L- and P-systems [13,14], gestural lexicons [15], knot theory [16], and mirror curves [17]. Of these, the work of Yanagisawa and Nagata [1], has similarities to this work. They begin with 5 rules for *kolam*, define square unit tiles that can be assembled into larger *kolam*s, define two types of nearest neighbor interactions between

doi:10.5047/forma.2015.005

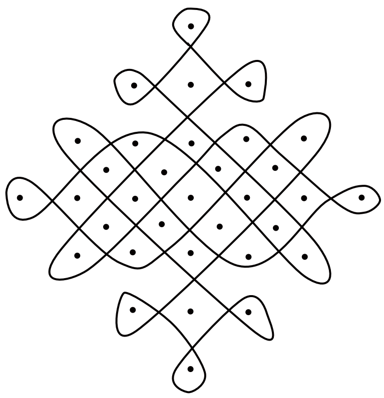

Fig. 1. Example of a *pulli kolam* called Brahma's knot.

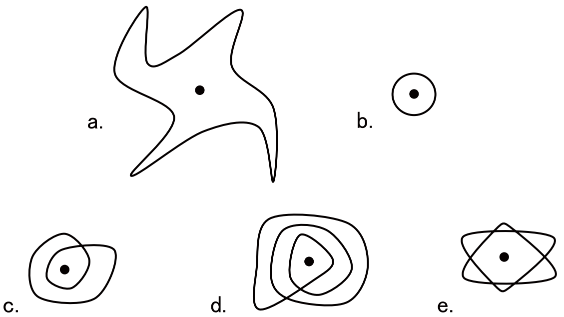

Fig. 2. Examples of *kolam*s around one dot that follow the mandatory rules **M**1–**M**3. An infinite number of *kolam*s are possible. Additional optional (**O**) rules, **O**1–**O**6, can limit the number allowed.

dots (line crossing, 1, or uncrossing, 0) and convert these tiles into binary number arrays. Nagata [18] also addressed the construction of a primitive *kolam* for an arbitrary dot array with a similar approach. In contrast, the work presented here has a purely topological approach: it defines only 3 mandatory rules for defining a *kolam*, has no standard tiles, generalizes the ideas to any arbitrary arrangement of dots arranged in any shape (not necessarily square arrays), generalizes to interactions between any two dots (instead of only the nearest or next nearest neighbors), and to three or more number of bonds between an interacting pair of dots. The work suggests that for a given number of dots, N, there are a limited number of parent *kolam* types from which all other *kolam*s originate. All parent *kolam*s within a parent *kolam* type are homotopic (or topologically equivalent).

2. How Many *kolam*s for One Dot ($N = 1$)?

Figure 2 depicts a single dot, and a variety of lines circumscribing it that follow the three mandatory rules mentioned above. The *kolam* in general could be amorphous in shape, as in Fig. 2a, and in the special case of Fig. 2b is a circle. Multiple circumscriptions around the dot are possible, as in Figs. 2c, d, and e.

It becomes immediately clear from Fig. 2 that the number of possible *kolam*s thus defined, with only the mandatory rules, is *infinite*. One may arbitrarily impose additional optional (**O**) rules to limit the number of *kolam*s. Here are some:

O1: Only one circumscription of the line is allowed around

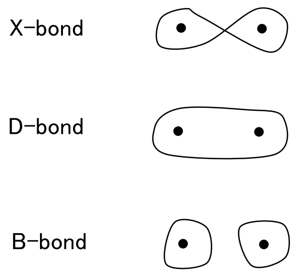

Fig. 3. Infinitely many types of bonds are possible between a pair of dots that follow the rules **M**1, **M**2, **M**3, and optional rules **O**1 and **O**2, three of which are shown here.

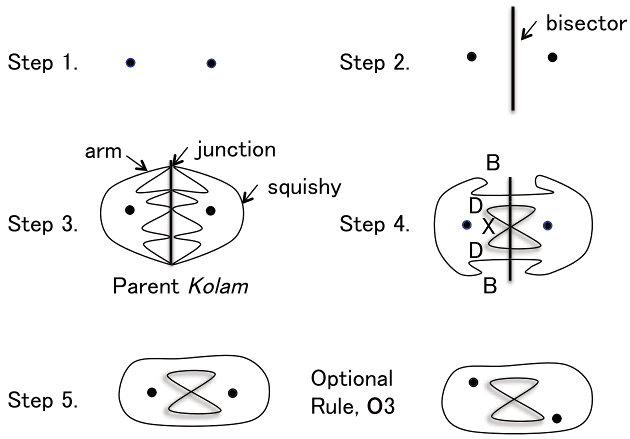

Fig. 4. Illustrating the construction of a *kolam* in 5 steps plus optional rule **O**3: The procedure is shown for $N = 2$ (2 dots) and $J = 5$ (5 junctions). If each junction is restricted have one of 3 types of bonds (X-, D-, or B-), it can lead to $3^5 = 243$ possible *kolam*s. One of these options, namely, B-D-X-D-B, is shown in the figure in Step 4. In the optional rule, the dots have been rearranged as an example of rule **O**3 after the *kolam* is drawn in Step 5.

each dot.

O2: A line circumscribing a dot should be as resourceful (simple) as possible, without additional unnecessary wiggles or flourishes (e.g. Fig. 2b is resourceful vs. Fig. 2a is not).

O3: While a *kolam* may be created by a minimum number of dots N needed for the 5-step method proposed below, one can then eliminate dots from, or add dots to, or move dots in a *kolam* after it has been drawn, provided the process does not violate the mandatory rules. The final *kolam* may thus appear to have N_{final} dots, where N_{final} may or may not be equal to N.

With **O**1 restriction, only 2a and 2b survive. With **O**1 and **O**2, only 2b will survive. Figure 2e, depicting a Star of David is a common *kolam*, which apparently is eliminated by **O**1. However, this *kolam* can also be generated by placing six dots ($N = 6$), one inside each ray of the star, and following the 5-step method proposed below. The 6 dots may later be erased, and one dot placed in the middle ($N_{final} = 1$) according to **O**3 to generate Fig. 2e. Another example is the Brahma's knot in Fig. 1, which can be generated by only $N = 25$ dots. However, Fig. 1 has $N_{final} = 33$ dots; the additional two horizontal rows of 4 dots each (total of 8 dots) in that *kolam* would be placed (according to

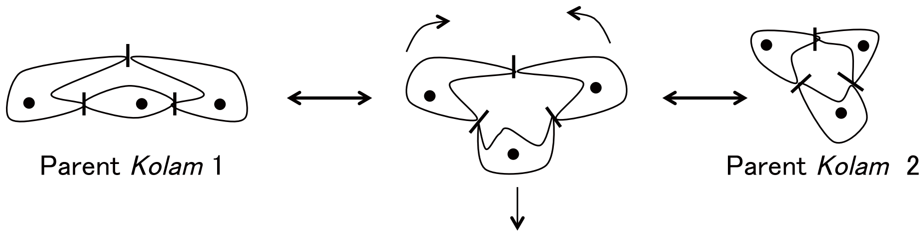

Fig. 5. Two possible parent *kolam*s for 3-dots ($N = 3$) and $J = 1$. The intermediate structure shows how one can distort parent 1 into parent 2, demonstrating that they are homotopic.

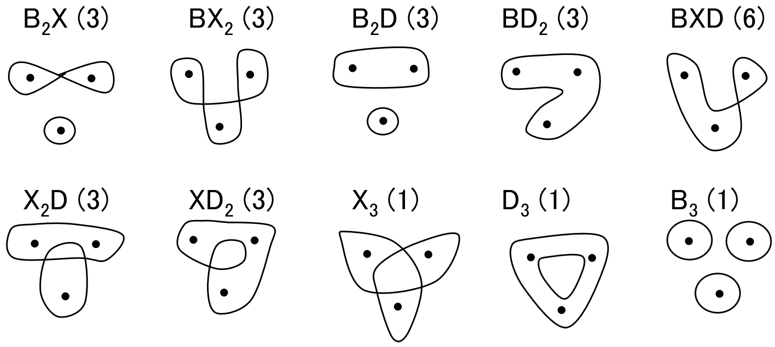

Fig. 6. The 27 *kolam*s generated from 3 dots ($N = 3$) and $J = 1$. There are 3 possible pairs of dots. The notation, B_2X (3), for example indicates that two of the pairs have broken-bonds and one pair has a cross-bond. The (3) in the end indicates that three such *kolam*s of the same type exist, generated by the permutation of the X-bond between the three pairs in the case of B_2X.

O3) after constructing the *kolam* with only 25 dots by the method proposed below.

3. Method to Construct *kolam*s for an Arbitrary Arrangement of *N* Dots

First, we define several types of bonds (b) between a pair of dots, as shown in Fig. 3. The X- and the B-bonds were discussed in Yanagisawa and Nagata [1] and they were indexed as a line crossing, 1, or an uncrossing, 0. The D-bond corresponds to additional variation (a type of two-dot joining, indexed as 2) over the pictorial code proposed by Nagata [18]. In general, there are infinitely many possible bond types but we will focus here only on the cross (X)-bond, the double (D)-bond, and the broken (B)-bonds ($b = 3$) in this work.

Next, we propose a 5-step method to build all possible *kolam*s for an *arbitrary* pattern of N dots in two-dimensions. These rules are illustrated for a simple 2-dot ($N = 2$) case in Fig. 4.

Step 1: Place the dots in *any* configuration of your choice in 2-dimensions.

Step 2: Draw a perpendicular bisector line segment between every pair of dots in the general case of following only rules **M**1–**M**3. (More generally, this line segment does not need to be a bisector, and does not need to lie between the two dots.) While the bisector is a line separating the two dots, the N-line used by Nagata [18] was used for connecting or bonding these dots. A line crossing on this N-line by Nagata [18] has analogies to the junction point on the bisector.

Step 3: Draw closed ghost-like figures around each dot, which we will playfully call *squishies*, suggesting that they are freely deformable. There will be N squishies for N dots. Each squishy will have J *arms* that touch a corresponding arm from a different squishy pairwise at the bisector line, leading to J *junctions*. We will call this structure, the parent *kolam*. All parent *kolam*s arising from different spatial arrangements of N dots can be classified into parent *kolam* types; within each type, all parents are topologically equivalent, or homotopic, as discussed further on.

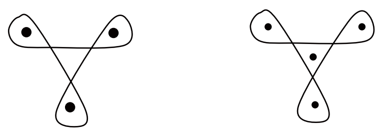

Fig. 7. The above *kolam* on the left might appear to be a 3 dot *kolam*. However, it is an $N = 4$ dot *kolam* (above right) created using the 5-step method described above. By erasing the center dot in the right *kolam*, one can generate the $N_{final} = 3$ *kolam* according to **O**3. These two *kolam*s are not homotopic.

Step 4: Now start drawing the *kolam* from any point on a squishy, and follow along until you reach a junction. Then transform that junction into a *cross-bond* (X-bond), a *double-bond* (D-bond), or a *broken-bond* (B-bond). Continue in a similar way until you return to the starting point. If some dots are still not encircled, start a new line from a squishy around one of those remaining dots, and continue till you return back to the start of that line. Repeat this process till the *kolam* is complete and all the dots are encircled.

Step 5: Smooth the curves so that the lines are resourceful according to **O**2. This will result in a *kolam* that will obey the rules **M**1, **M**2, and **M**3.

As an optional rule, you can eliminate any or all dots, or add new dots, or move the existing dots according to **O**3. In addition, one may impose further optional rules to whittle down the number of *kolam*s:

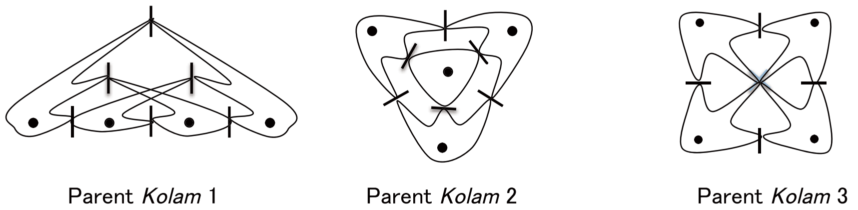

Fig. 8. Three types of parent *kolam*s for $N = 4$ dots and $J = 1$. Parents 1 and 3 are homotopic and form one parent type. Parent 2 forms a second parent type.

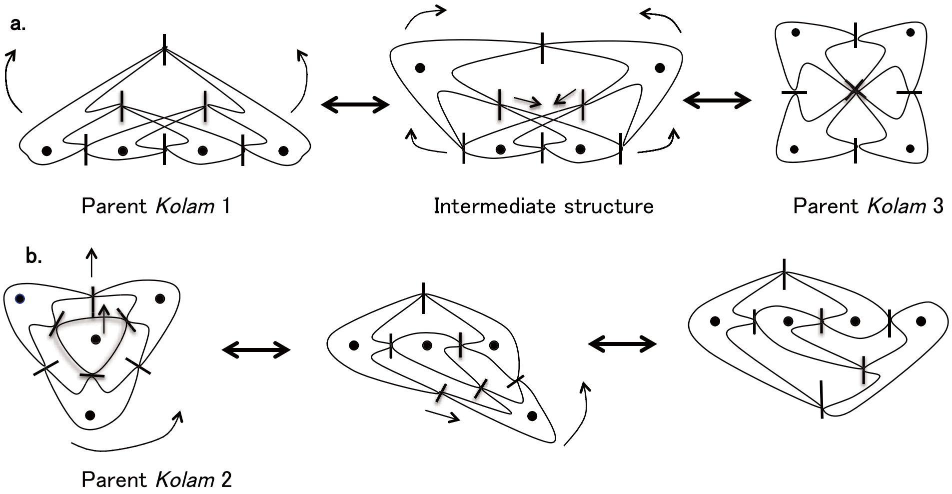

Fig. 9. Parent *kolam*s 1 and 3 for $N = 4$ and $J = 1$ shown in Fig. 8 are demonstrated to be topologically equivalent by continuously deforming parent 1 into 3 in panel (a); hence they form a single parent type. Panel (b) shows that distorting parent 2 in Fig. 8 does not lead to parent 1; hence they are distinct parent types.

O4: Only the nearest neighbor dots interact through bonds other than broken bonds. All other bonds are broken.
O5: Only one junction ($J = 1$) is allowed between one pair of dots.
O6: Symmetry equivalent junctions in the parent *kolam* will have the same type of bonds. To find sets of symmetry equivalent junctions, visual inspection of possible rotations axes and mirror symmetries is recommended. For a mathematical approach, find the point group of the arrangement of dots, and using the symmetry operations of the point group, see which set of junctions transform into each other.

In general, with rules **M**1, **M**2, **M**3 and optional rules **O**1 and **O**2 in place, with J number of junctions per pair of dots, N and with b types of bonds allowed (Fig. 3), one can write the number, K, of possible *kolam*s as

$$\#Kolams = K = b^{JN(N-1)/2}, \qquad (1)$$

where the exponent of b is the number of possible junctions between all possible pairs of dots. For example, if $N = 2$ (2 dots), $J = 1$ (1 junction) and $b = 3$ (3 bonds), then $K = 3$. These 3 *kolam*s are shown in Fig. 3. Obviously, K gets large very quickly as J, b and N increase. In the rest of this work, we will restrict ourselves to $J = 1$ and $b = 3$.

If the optional rule **O**6 is imposed in addition, and symmetry equivalent junctions identified, let there be g groups, each containing S_g number of symmetry equivalent junctions, such that $\sum_g S_g = JN(N-1)/2$. Then the number of possible *Kolams* (Eq. (1)) can be revised as $K = b^g$.

Note that we assert in Step 5 that this procedure will always result in a *kolam* that obeys the mandatory rules. This arises from the rules of construction. The parent *kolam* is always drawn in the above steps in such a way as to not violate the three mandatory (**M**) rules: all dots are circumscribed by squishies and there are no loose ends in the parent *kolam*. Nor does the transformation of the junctions in Step 3 violate these rules: the bonds where lines cross, e.g. the X-bond, cross at a single point per crossing. Hence the final *kolam* also follows the minimal mandatory rules **M**1–**M**3.

4. Exploring *Kolam*s with 3 Dots ($N = 3$)

The number of possible *kolam*s for $N = 3$ following rules **M**1–**M**3 and optional rules **O**1, **O**2, and **O**5 ($J = 1$) can be computed from Eq. (1) as $K = 3^{1\times 3\times(3-1)/2} = 3^3 = 27$. Two different parent *kolam*s for $N = 3$ are shown in Fig. 5.

Parent 1 places the three dots on a line, while parent 2 places them in a triangle. These two parent *kolam*s are topologically equivalent, or *homotopic*. In other words, a continuous distortion of one structure can result in the other without cutting or breaking bonds, as shown by a transformation through the intermediate structure in Fig. 5. Hence, every one of the 27 *kolam*s derived from parent *kolam* 1 will have a topologically equivalent cousin *kolam* derived from parent *kolam* 2. Thus we can conclude that for $J = 1$, all $N = 3$ *kolam*s arise from a single parent *kolam* type.

The 27 *kolam*s derived from parent *kolam* 2, with the

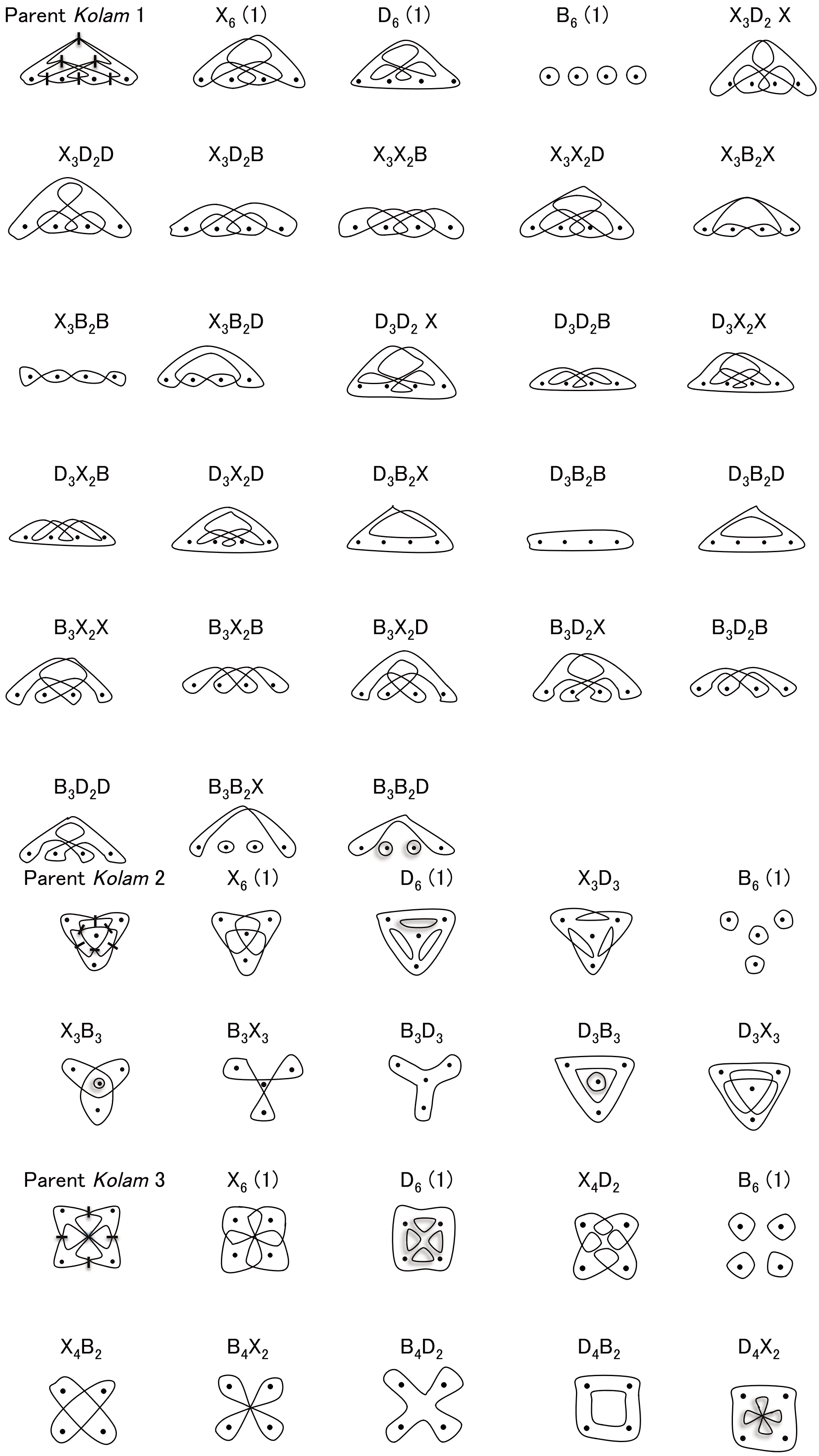

Fig. 10. Four dot ($N = 4$) *kolam*s derived from the three parents in Fig. 8, under the rules of the rules **M**1–**M**3, and optional rules **O**1 (only one circumscription per dot), **O**2 (simplifying the line), **O**5 ($J = 1$), and **O**6 (symmetry equivalent junctions will have the same type of bond). Note that the parent *kolam*s in this figure have been chosen in the special shapes of a line (parent 1), an equilateral triangle (parent 2) and a square (parent 3). These choices as well as the optional rules eliminate many *kolam*s that a reader might otherwise be able to visualize.

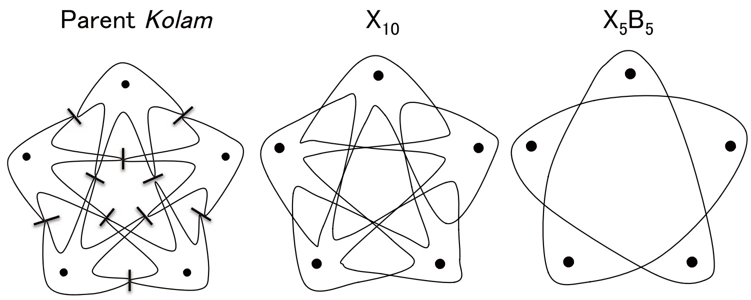

Fig. 11. An example of a parent *kolam* and two children *kolam*s for $N = 5$ and $J = 1$.

special case of the 3 dots arranged in an equilateral triangle, are shown in Fig. 6. Did we find all possible *kolam*s with $N = 3$? If so, how about the *kolam* on the left in Fig. 7? It turns out that this *kolam* is captured by the proposed method for $N = 4$, where an additional dot is placed in the middle of Fig. 7. This is discussed in the next section. The example is again illustrative of the fact that a *kolam*, once created, is distinctive in its own right, irrespective of the presence or absence of dots. *The characteristic N for a given kolam may be defined as the minimum number of dots required for generating the kolam with the above 5-step method*. However, note that when dots are removed or added to a *kolam*, the resultant *kolam*s may no longer be topologically equivalent to the original *kolam*.

5. Exploring *Kolams* with 4 Dots ($N = 4$)

Three different configurations of parent *kolam*s are shown in Fig. 8 for $N = 4$.

It is possible to show that parents 1 and 3 are homotopic. Such equivalence is shown in Fig. 9a, and hence they form a single parent type. However, parent 2 forms a distinct parent type as shown in Fig. 9b since parent *kolam*s 1 and 2 cannot be distorted into each other without the lines crossing over the dots in two dimensions. The number of possible *kolam*s for any parent *kolam* with $N = 4$ following rules **M**1–**M**3 and **O**1, **O**2, and **O**5 can be computed from Eq. (1) as $K = 3^{1\times 4\times(4-1)/2} = 3^6 = 729$.

The 729 possible *kolam*s from each parent is a large number, and so we choose here to impose additional restrictions in order to explore only a subset. For example, optional rule **O**6 suggests that symmetry equivalent junctions will have the same type of bond.

This allows for the symmetry of the parent phase to be preserved while bonds are formed. The various *kolam*s derived from three different parent *kolam*s (1, 2, and 3) in Fig. 8 under the rules of **M**1–**M**3 and **O**1, **O**2, **O**5, and **O**6 are shown in Fig. 10. For parent *Kolam* 1 in Fig. 10, there are 3 groups ($g = 3$) of symmetry equivalent junctions related by a vertical mirror symmetry. Thus the number of *Kolam*s with $J = 1$ is $K = 3^3 = 27$. For both the special cases of parent *Kolam* 2 (dots forming an equilateral triangle) and *Kolam* 3 (dots forming a square), $g = 2$ arising from a 3-fold and 4-fold rotational axes respectively, and hence $K = 3^2 = 9$ as shown. We note that B_3X_3 with $N = 4$ captures the *kolam* that was missed in Fig. 7 by $N = 3$.

6. Conclusions

We have demonstrated a method of generating countless *kolam*s from user-defined dot arrangement on a surface. This method can be mastered by anyone without the need to understand the detailed mathematics behind *kolam*s. For a give number, N, of dots in any spatial arrangement on a surface, the number of possible *kolam*s that follow only the mandatory rules **M**1–**M**3 is infinite, even for a 1-dot *kolam* ($N = 1$). However, by following additional optional rules **O**1 and **O**2, this number is finite as given by Eq. 1. Addition of rule **O**6 modifies this equation.

We show by example that for a given number of dots N, a set of parent *kolam* types exist, from which all possible *kolam*s can be generated. All parent *kolam*s within a single type are homotopic. Hence the resultant *kolam*s from these homotopic parents will also have corresponding homotopic cousins. Though a rigorous proof for such homotopy in general has not been presented, it can be argued based on the method of construction similar to that shown in Fig. 9.

*Kolam*s with higher N get richer and more complicated quickly. For example, Fig. 11 shows an example parent *kolam* for $N = 5$ and $J = 1$, and two possible children *kolam* arising from it. The readers are encouraged to try generating other parent and children *kolam*s for this case.

There are several advantages to this simple method:

(1) It is applicable for any number of dots, N.

(2) The dots can be arranged in any configuration in 2-dimensions.

(3) While the proposed method may not always guarantee aesthetics, it is simple enough for a user to impose additional aesthetically appropriate optional rules.

(4) A computer program can vary b, J, and N for generating numerous *kolam*s following the three mandatory rules, plus any number of user-defined optional rules.

This leads to the possibility of creating an interactive website or a mobile app that can help a user to generate *kolam*s at will. Such an app will get the user involved in the creative process, including young children who may be introduced to art, symmetry and topology through *kolam*s. The method is also applicable to generating other similar patterns such as some of the Chinese and Celtic knots.

Acknowledgments. The authors would like to thank Mikael Rechtsman and Chaoxing Liu from the Pennsylvania State University for exciting and valuable discussions. The ideas for this work tangentially originated from developing the theory of distortions funded by the National Science Foundation grant number DMR-1420620 and DMR-1210588.

References

[1] Yanagisawa, K. and Nagata, S. (2007) Fundamental study on design system of kolam pattern, *Forma*, **22**, 31–46.

[2] Siromoney, G., Siromoney, R. and Krithivasan, K. (1974) Array grammars and kolam, *Computer Graphics and Image Processing*, **3**, 63–82.

[3] Siromoney, G. and Siromoney, R. (1987) Rosenfeld's cycle grammars and kolam, graph-grammars and their application to computer science, *Lecture Notes in Computer Science*, **291**, 564–579.

[4] Subramanian, K. and Siromoney, R. (1987) On array grammars and languages, *Cybernetics and Systems*, **18**(1), 77–98.

[5] David, N. G., Robinson, T., Thomas, D. G., Balamurugan, B. J. and Samuel, S. C. (2007) Graph grammers for kolam patterns and honeycomb patterns, *Int. J. Mathematical Sci.*, **6**, 355–367.

[6] Pradella, M., Cherubini, A. and Crespi Reghizzi, S. (2011) A unifying approach to picture grammers, *Information and Computation*, **209**, 1246–1267.

[7] Siromoney, R., Subramanian, K. G. and Rangarajan, K. (1976) Control on kolam arrays, *Information and Contorol*, **32**, 272–275.

[8] Siromoney, R., Rajkumar Dare, V. and Subramanian, K. G. (1983) Infinite arrays and infinite computations, *Theotetical Computer Science*, **24**, 195–205.

[9] Demaine, E. D., Demaine, M. L., Taslakian, P. and Toussaint, G. T. (2007) Sand drawings and Gaussian graphs, *Journal of Mathematics and the Arts*, **1**(2), 125–132.

[10] Subramanian, K. G. (1979) A note on regular kolam array grammers generating right triangles, *Pattern Recognition*, **11**, 343–345.

[11] Robinson, T. (2007) Extended pasting scheme for kolam pattern generation, *Forma*, **22**, 55–64.

[12] Allouche, P. G., Allouche, J.-P. and Shallit, J. (2006) Kolam indiens, dessins sur le sable aux iles Vanuatu, courbe de sierpinski et morphismes de monoide, *Amm. Inst. Grenoble*, **56**(7), 2115–2130.

[13] Prusinkiewicz, P. and Hanan, J. (1989) *Lindenmayer Systems, Fractals, and Plants*, Springer-Verlag, New York, ISBN 0-387-97092-4, *Lecture Notes in Biomathematics*, **79**.

[14] Subramanian, K., Saravanan, R. and Robinson, T. (2007) P systems for array generation and application to kolam patterns, *Forma*, **22**, 47–54.

[15] Waring, T. M. (2012) Sequential encoding of tamil kolam patterns, *Forma*, **27**, 83–92.

[16] Ishimoto, Y. (2007) Solving infinite kolam in knot theory, *Forma*, **22**, 15–30.

[17] Gerdes, P. (1999) On Lunda-designs and some of their symmetries, *Visual Mathematics* (ISSN 1821–1437), **1**, 1. http://symmetry-us.com/journals/paulus/index.html

[18] Nagata, S. (2007) Digitization and analysis of traditional cycle patterns in the world, and their contemporary applications, *Forma*, **22**, 119–126.