origami draft 1

Benjamin Wettle

March 2020

1 Introduction

This Algorithm uses box pleating methods to design optimal crease patterns for any given origami design.

2 Background

Origami is the ancient art of folding paper to represent various forms. Traditional artists found step by step instructions to fold the base, and then add details. These designs were found by experimentation, and modification of existing designs. More modern artists such as Robert Lang [1] have developed algorithms to generalize this process to work for arbitrary bases.

2.1 Definitions

TODO define Nodes, leaf nodes, river nodes, bases, and efficiency.

2.2 problems with circle packing

circles represent the minimum space that a node must take up in the optimized design. of course, any larger shape that completely contains the circle could be used. One common alternative to circles is to use squares instead. though this may reduce the efficiency, there are other added benefits.

First, the wasted space is only potential. It is highly unlikely that all or even most of the "corner" space would be used in a circle packed design. figure(1) shows a simple origami base. figure 2 shows designs with circle packing and box packing on the right and left respectively. the loss in efficiency is far from the theoretical maximum, an is out weight by concerns outside of effiecny.

folding time for many circle packed designs, the locations of creases are arbitrary, and not easy to fold in practice.tools like Reference finder can find points and lines through simple folds. this adds extra layers of complexity and can clutter designs with excessive creases. with a box packed design, the folds are

all at multiples of 45 from each other. this is easily fold able, and requires no extra tools to fold. in addition, the flaps produced are standardized. with a circle packed design, a flap could have a range of possible angles and thicknesses, all or which impact the creative versatility of the flap. with a box pleated design, detailing can be experimented with before the full base is done. this standardization also makes the actual folding process easier and less open to error.

3 Approach

the core of the algorithm is implemented int the layout and skeleton classes. once the origami design is ready to be optimized, a layout is created. this uses all the information about the design, stored in the paper class. For constant evaluation, it is unnecessary to handle all of this information. only the bare minimum is stored in the layout. this consists of: a list of only the leaf nodes, the minimum distances they have to be from each other, and any conditions (such as optimizing to a rectangle , or similar).

then comes the core question of the optimization- where is the best place for these nodes to go? each potential arrangement is stored in the skeleton class, and a list of all the skeletons is stored in the layout. each node will be added sequentially, so in all there will be n steps, where n is the number of leaf nodes. The first node is easy to place, since all distances are relative. it can be placed at any point, such as 0,0, or wherever it already is. since optimizing a single node is trivial, we must also account for the rest of the nodes. the second node could go many places, as long as it meats the minimum required distance. at this point, we simple don't know were the best spot is, so we will place it in several spots. A square with sides of 2*(minimum distance+buffer) is searched for potential locations. initially this buffer is set to zero, in the hope that the node can be found close by, without many gaps in the design. of the spaces, those overlapping the first node can be automatically eliminated. then, we check to make sure the new node does not overlap any other nodes. if it does not, we know that is a possible design. most likely, it is not the best design, as tons of designs are both possible, and highly wasteful. a simple sorting method compares the sizes of all the skeletons, to see which are the most wasteful so far. it is possible that the smallest design at this stage is worse in later steps, so many of them are kept. the limit is set by default to 5,000. any additional designs are discarded. once this short list is made, it is time to add the next node. for each skeleton , the previous steps are repeated. potential locations are searched, the possible skeletons are sorted, and only the best are kept. this is repeated until all leaf nodes are added. the most difficult step in the process is the sorting step. the skeletons are sorted with a merge sort, which sorts with an order of m*log(m), where m is the number of skeletons. This number is $\max_{p}opulation*4*((*minimumdistance+buffer)^2-minimumdistance)^2).withoutsortingandselectingthebestskeletons, thenumberofskeletonsconsideredwouldg$

$Area * log(Area), where area is the effective area searched in each step.$

4 Evaluation

TODO compare time and efficiency of a few designs to tree maker.

5 Related work

mention robert lang, tree maker, reference finder, and origami design secrets .

6 Conclusion