Finding Reference Points by Folding

By Robert J. Lang

When one is designing an origami model using mathematical tools such as *TreeMaker*, the crease pattern is usually defined mathematically, rather than by folding. Important points in the pattern — junctions of several creases, which are called *reference points* of the crease pattern — are commonly specified entirely numerically, rather than as a result of a series of folds. To fold such patterns, one can record the coordinates of the important points and then compute, measure and mark their location on the paper to be folded. But making marks on the paper is awkward and inelegant; it is inconvenient to require a ruler and calculator to fold origami! There is an aesthetic benefit to being able to start with an unmarked square and proceed to the completed model entirely by folding. To do this, we need a way of finding reference points by folding alone when all we have is an algebraic or numerical description of the point.

This challenge of finding folding sequences for the location of a reference point given its mathematical definition is a problem of both mathematical and practical interest, and it has seen considerable progress in recent years. Finding reference points is closely related to the concept of geometric construction — a field familiar to anyone who has manipulated compass and straightedge in high-school geometry. The field of origami constructions is considerably richer than compass-and-straightedge constructions: a partial listing of some of the accomplishments and discoveries in this field includes:

— Construction of any rational fractional distance (expressed as a fraction of the side of the unit square) with algorithms by Husimi, Fujimoto, Noma, Haga, Mosely, and me;

— Construction of any binary rational fraction (a fraction whose denominator is a perfect power of 2);

— How to trisect an arbitrary angle, with constructions by Justin and Abe;

— Construction of cube roots, including a marvelous construction of the cube root of 2 by Peter Messer;

— Solutions of general quadratic and cubic equations and construction of perfect N-gons for broad classes of N by Robert Geretschläger (including all N up to 20 except N=11).

Using these and related techniques, it is mathematically possible to construct by folding alone any point whose coordinates in the unit square are given by any combination of integers, simple arithmetic (+, –, *, /), and square and cube roots.

However, these constructions are not always useful in a practical sense. The mathematical origamist looks at the problem of origami geometric construction as an intellectual exercise where mathematical exactness is an absolute requirement and the construction of extraneous or leftover creases is of little or no consideration. To the practical origami designer, however, many of the mathematical constructions described above leave the paper littered with extraneous creases even for the construction of a single point. Furthermore, these constructions can be very difficult to do precisely, involving the transfer of distances from one location on the paper to another and/or locating points by finding the intersection of two creases at very shallow angles. Paradoxically, a mathematically exact folding sequence can lead to a very imprecise result when one actually applies hands to paper.

Furthermore, mathematical exactitude is not required for most practical purposes, and is, in fact, utterly unachievable, since paper is an imperfect medium and there is uncertainty inherent in the process of folding itself. There is no point in using a mathematical folding sequence that has theoretically infinite accuracy if either the needed or the attainable accuracy is only some finite value.

And in fact, I have found empirically that an accuracy of one part in 1000 is sufficient to fold any model and it is difficult, if not impossible, to do better than this when folding. For a standard 25 cm square, 1 part in 1000 is a quarter-millimeter — about three hair diameters! In many cases, 1 part in 100 accuracy will suffice.

It is also very important to the origami designer that unnecessary creases be minimized or eliminated and those that remain should be small and unobtrusive. If the folder must make 100 sequential creases to locate a single reference point, or the paper must be covered with creases that serve as intermediate reference points — the folding sequence, the finished model, or both, will be aesthetically unpleasing.

And so, a very practical problem faced by the origami designer can be stated quite succintly, albeit imprecisely: how can you locate an arbitrary point on a unit square (1) by folding alone, (2) to reasonable accuracy, (3) with as few folds as possible, (4) leaving as few creases on the paper as possible?

This statement is, to a mathematician, an ill-defined problem. How accurate is “reasonable accuracy?” How few is “as few as possible?” There are some tradeoffs. In general, higher accuracy requires more creases. We can, however, place some quantitative limits on these tradeoffs. I have previously looked at the problem of finding a reference point along the edge of the paper. The binary folding method† lets you construct an approximation to any point along an edge with a well-defined accuracy; a sequence of *N* folds can be found with a maximum error of $2^{-(N+1)}$. Thus, for example, an accuracy of 1 part in 256 — a maximum error of .004 — can be obtained with no more than 7 creases for any location on the edge of the square. (Of course, some distances can be found with fewer creases, but the worst-case scenario is 7 creases.)

A nice feature of the binary folding sequence is that all intermediate marks take the form of small pinches which can be made arbitrarily short. There are no extended creases that run across the square that potentially mar the surface of the finished model, and so it is reasonable when searching for alternate algorithms to add the restriction that the only creases allowed are short pinches.

The binary folding sequence utilizes pinches to form intermediate marks along a single edge of the square — the same edge that the desired reference point is located upon. More recently, I’ve looked at the problem of locating a point along an edge of the square where the intermediate marks can be made along *any* of the four edges. I did this by simply looking at all possible marks that can be made recursively by bringing one mark on any edge to another mark on any edge, starting with the four corners of the square as the original four “marks.” Remarkably, it turns out that it is possible to locate any point to an accuracy of better than 1 part in 100 with no more than 4 creases — whereas the binary sequence would require up to 6 creases for the same level of accuracy.

† To fold a proportion *x*, write *x* in binary, *e.g.*, *x*=.100111. Then, beginning from the *right* side of the fraction, for every 1, fold the top down to the last crease made and for every 0, fold the bottom up to the last crease. The final crease divides the edge in the desired proportion.

The problem changes when you start to consider that there are usually multiple reference points that need to be found, as the amount of precreasing required can grow significantly. Even with 4 creases per reference point, if there are 10 reference points to be found, there could be as many as 40 folds required to find those 10 points. That's a lot of precreasing! The number of folds can be reduced significantly if several of the reference points can share earlier creases, or better yet, if all reference points can be derived from a small set of starting marks.

One set of starting marks that is fairly easy to find is the set of marks spaced 1/8 of the side of the square around the outside of the square (see figure). Each of these marks individually can be made with no more than 3 creases, and the entire set of 28 crease marks can be constructed with 14 creases (making pinches at each end of each crease). With the addition of the original 4 corners of the square as marks, the set of 1/8 marks gives 32 points around the outside of the square that can be used to find other points along the edge.

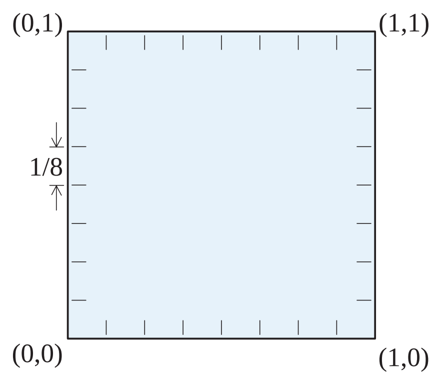

Figure 1. A square showing 1/8 marks around the edges.

It turns out that, starting with these 32 marks, you can locate any other point on any edge of the square to an accuracy better than 1 part in 100 with only a single additional crease! Figure 2 shows an example, for how we find the point (1/π,0)=(0.3183,0). A single crease gives the point (0.3125,0), which has an error of 0.0058 — plenty accurate for most folding purposes.

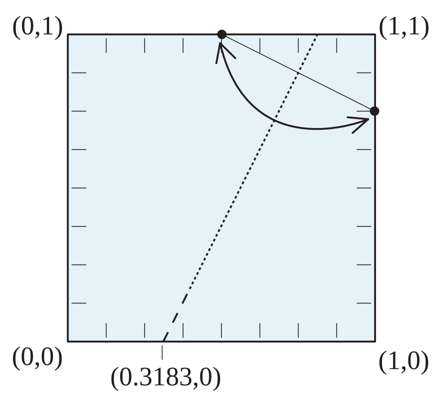

Figure 2. A folding sequence that gives the point (1/π,0) to an accuracy of 0.5%.

All of these creases are generated by bringing one mark to another mark and making a pinch on the edge of the paper. Unfortunately, there is no simple way to deduce the appropriate combination of marks for a given reference point from the coordinates of the reference point. However, it is a straightforward computer programming task to simply enumerate all possibilities and pick out the combination that comes closest to the target.

Let's compare these last two approaches for a set of *N* marks. If we find a sequence for each mark individually, then the odds are that the *N* sequences will not share any intermediate marks; thus, the total number of creases to be made is approximately 4*N*. On the other hand, if we start with the 1/8 marks as a starting point, we must make 14 creases before we begin, but each mark will only require a single additional crease; the total number of creases is (14+*N*). The crossover point between these two approaches occurs when $4N \approx 14+N$, or $N \approx 5$. For fewer than 5 marks, it's generally more efficient to find each mark individually; for more than 5, it's generally more efficient to start with the 1/8 points as reference points. However, since the possible marks are distributed irregularly around the outside of the square and certain marks may have fortuitously elegant sequences, it's a good idea to check both approaches.

That's all well and good, but not all reference points are located on the edges of the paper; one commonly has to find a reference point somewhere in the interior of the paper — which brings me to the heart of this discussion. Like the mariners of old, we have, to date, hugged the shoreline of the great ocean of the unit square. But we now have the tools to strike out across the vast expanse of uncharted, featureless paper.

The first step is to decide how we will locate a point in the interior of the square. We located a point on the edge by the intersection of a short crease — a pinch — with the edge. We can similarly locate a point in the interior by the intersection of two pinches.

But how are pinches defined? Let us first look at pinches constructed from marks around the edge of the paper, like the set of 1/8 marks. A pinch is a short segment of a crease. There are two fundamentally different ways of constructing creases from edge marks. First, we can bring one mark to another and flatten the paper, making a crease; this crease is the perpendicular bisector of the line between the two marks. The second way is to make a crease that itself runs between the two marks. I'll call a crease of the first type a "bring" crease and a crease of the second type a "connect" crease.(Interestingly, when teaching origami to novices, they learn how to make a "bring" crease very quickly, but commonly have a great deal of trouble making a "connect" type crease.)

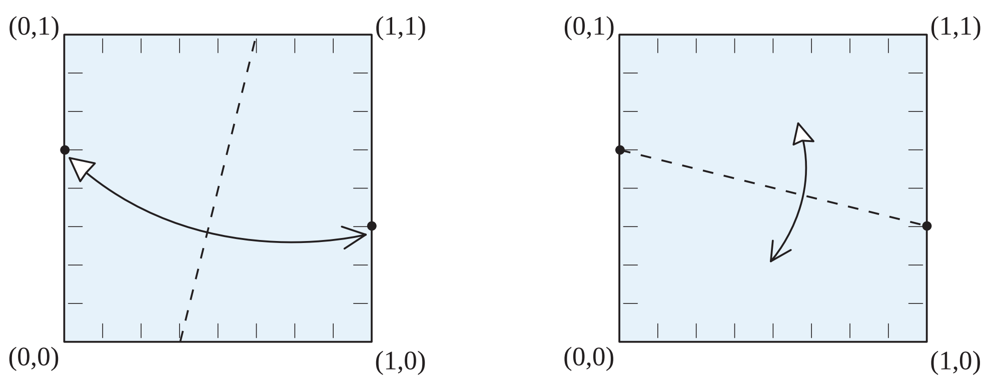

Figure 3. Left: a "bring" crease. Right: a "connect" crease.

So, suppose we start with the 32 marks spaced by 1/8 around the outside of the square. Every pair of marks can be used to construct either a "bring" crease or a "connect" crease. So from these 32 starting marks, there are 32*31/2=496 possible pairs, 2 types of crease for each pair, so a total of 992 possible creases that can be formed.

They are not all distinct (different) creases, however. Figure 4 shows two different mark pairs that produce exactly the same crease line. There are many such pairs. If we eliminate duplicates, we find that there are 640 *distinct* crease lines that can be formed from those 32 marks.

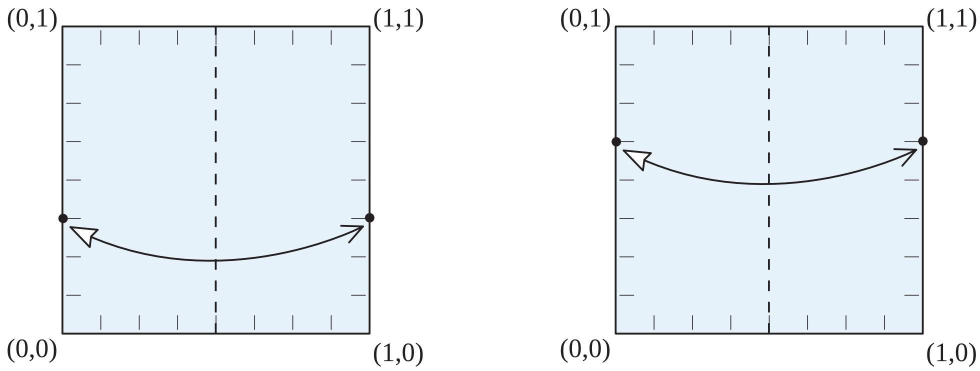

Figure 4. Two different mark pairs that produce the same type of "bring" crease.

Now, every pair of these 640 lines has an intersection somewhere (if the lines aren't parallel, that is). So potentially, there are 640*639/2 line intersections, which means that those 640 lines can potentially generate 204,480 marks, each defined by the intersection of a pair of lines. If those marks were distributed uniformly within the unit square, the area per mark is 1/204,480, and the average distance between marks is 1/√204,480, or about 1/452. So, in principle, for any point within the unit square, there is some mark fairly close by, located, on average, about .0012 units away.

However, not all of these marks fall within the unit square. Some marks fall outside of the square; of course they are useless for the purpose of locating reference points *inside* the square. Other marks may be defined by the intersection of two lines that are nearly parallel, which, as a practical matter, is prone to small errors in folding and should be avoided. If we discard marks outside the square and discard marks formed by lines at less than 30° to one another, it turns out that we are left with 111,479 marks in the unit square. This still gives an average distance between marks of .003, or an average error of .0015 — fine for all practical purposes!

Well, that's still not quite right. Although the *average* error is .0015, this value is only attainable if the marks are perfectly evenly distributed across the square, which they are not; their distribution is somewhat irregular (and in fact, even though we eliminated duplicate lines, there are still some duplicate marks). To truly ascertain the utility of this set of marks, we need to look at some statistical properties. Although we cannot examine every possible point in the square, a large random sampling should give us a general idea.

For 1000 randomly chosen test points, I calculated the distance to the closest marks and sorted the distances in order of their size. The results are as follows:

Percentile	Maximum error
10th percentile	0.000677666
20th percentile	0.00103332
50th percentile	0.0021392
80th percentile	0.00385641
90th percentile	0.00511358
95th percentile	0.00631435
99th percentile	0.0103066

This is still very nice! Half of the points in the square are within .002 of a mark. For over 95% of the possible reference points in the square, the closest mark is within .006 of the point, and for 99% of all reference points, there is a mark within .01 of the point.

So, in addition to generating edge reference points, the 1/8 marks around the edge of the square can also be used to produce interior reference points with very good accuracy. Each reference point requires 2 creases (1 for each pinch); thus, if a crease pattern has N_e edge reference points and N_i interior reference points, the total number of creases required is $14+N_e+2*N_i$.

Interestingly, if we double the number of starting points (say, by starting with the 1/16 points around the edge), we quadruple the number of possible creases, and increase the number of potential marks in the interior of the paper by a factor of 16. This increases the density of marks by a factor of 4, and so will decrease the average error also by a factor of 4 as well.

As was the case with the location of edge marks, there is a problem: there is no simple way to figure out from the coordinates of a target reference point what specific pinches are required to find the closest mark. Fortunately, with the advent of modern computers, there is no need for a simple method if brute force will do the trick — and in this case, it will. It is a straightforward task to program a computer to construct all 111,479 marks; then given a target reference point, one can sort through the list and find and print out the closest mark.

I have written a small C++ program called *ReferenceFinder* that does this. It takes as input the coordinates of a target reference and prints out the 10 best folding sequences for locating that point, utilizing the 1/8 marks around the outside of the square as starting reference points. The program is small and simple (it has a text-only command-line interface), and I have compiled both Mac and Windows versions. You can find both executables (and the source code, if you want to compile it for another platform or are simply interested in seeing how it was done) in the rugcis origami archives (ftp://rugcis.rug.nl).