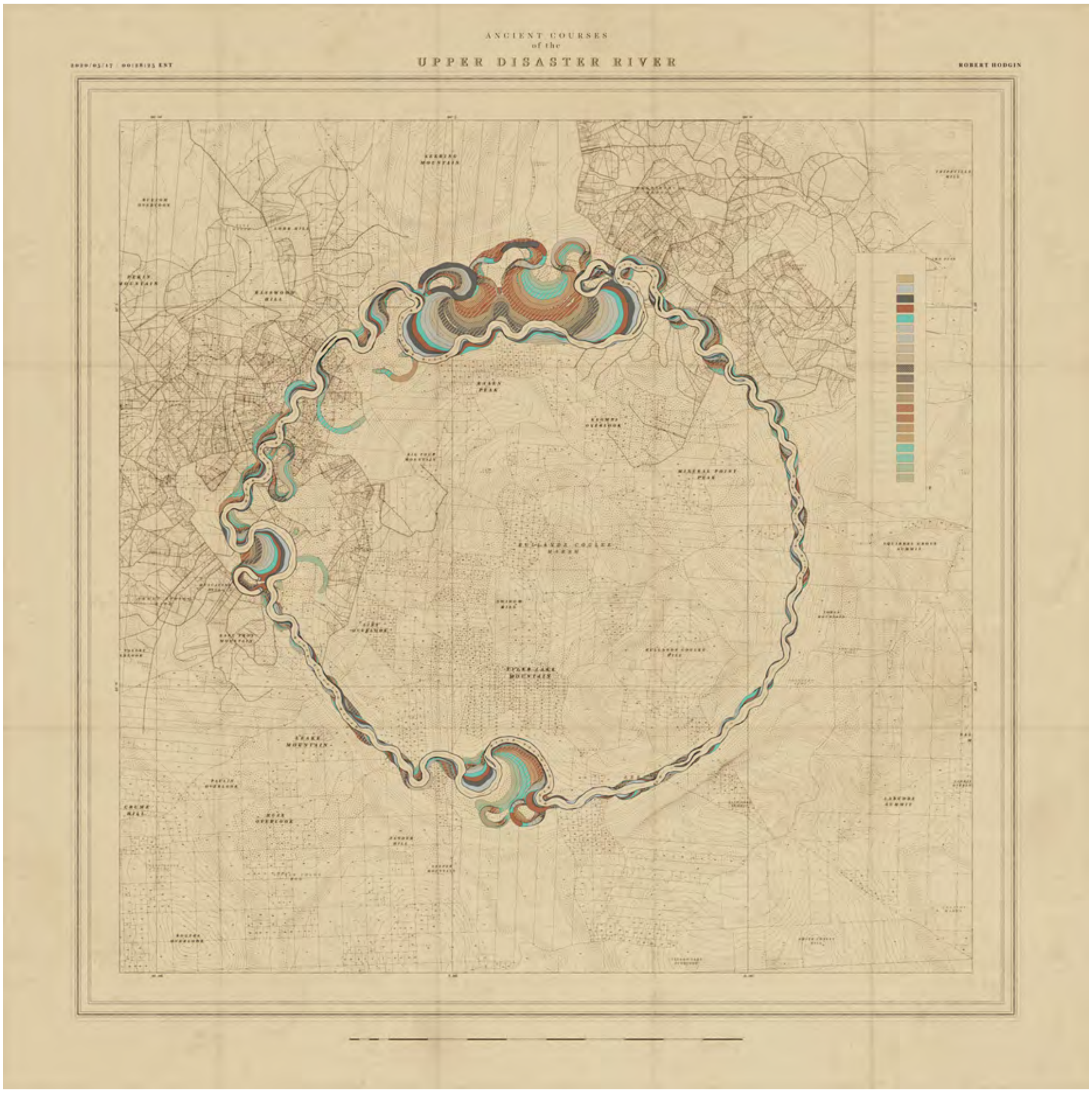
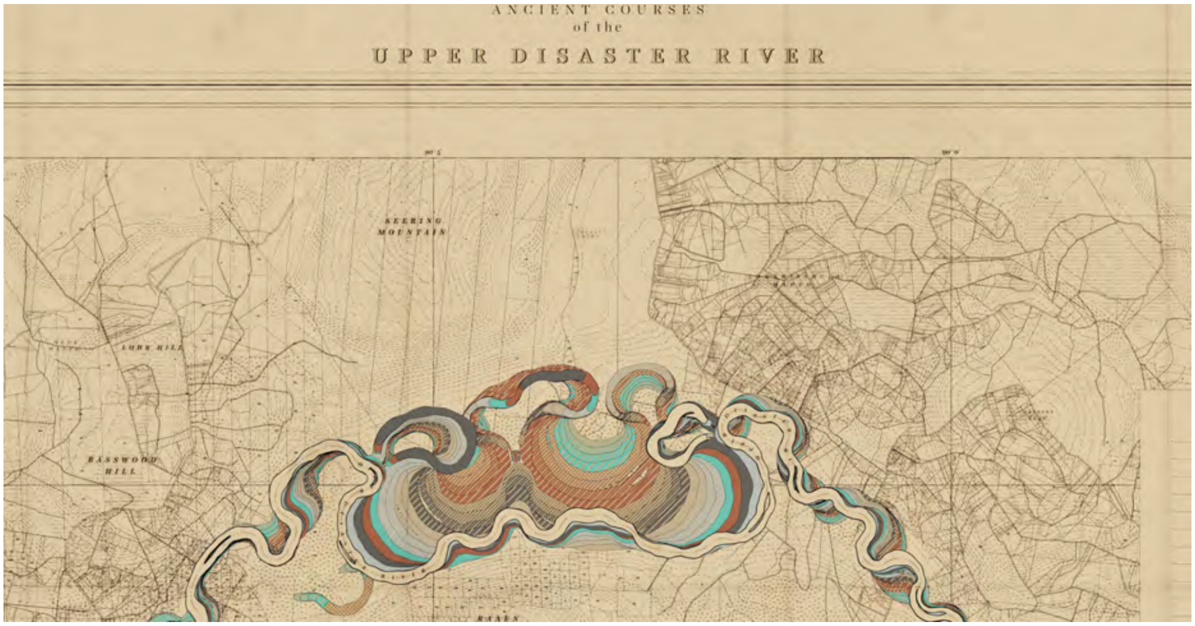
Robert Hodgin

May 2020

Meander is a procedural system for generating historical maps of rivers that never existed.

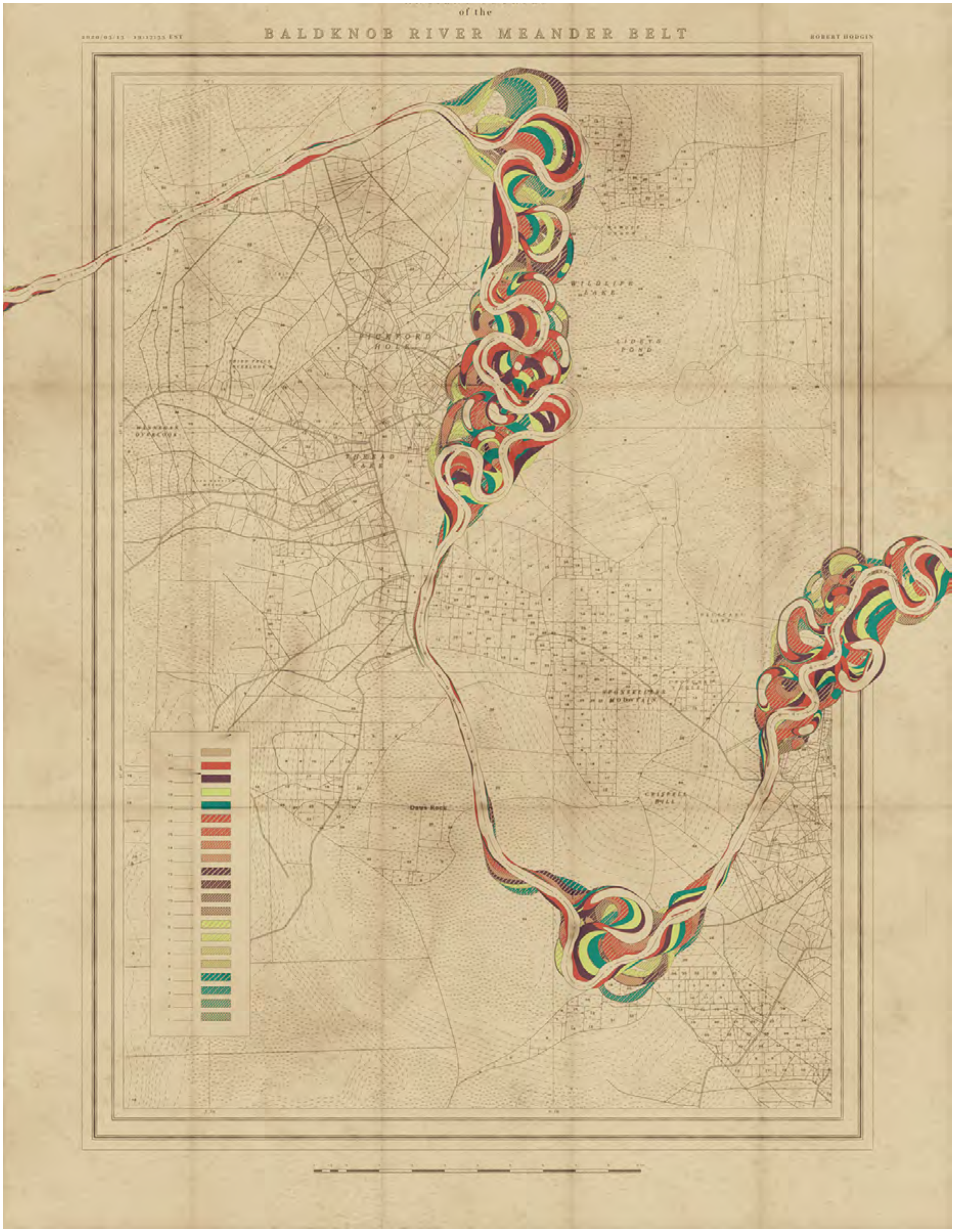


Robert Hodgin

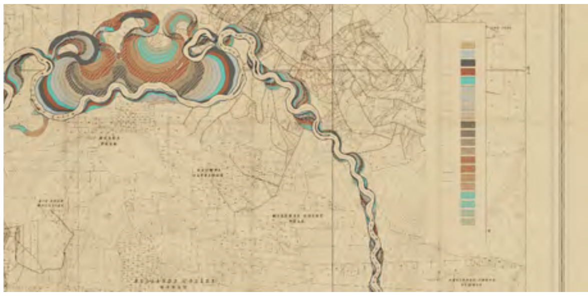
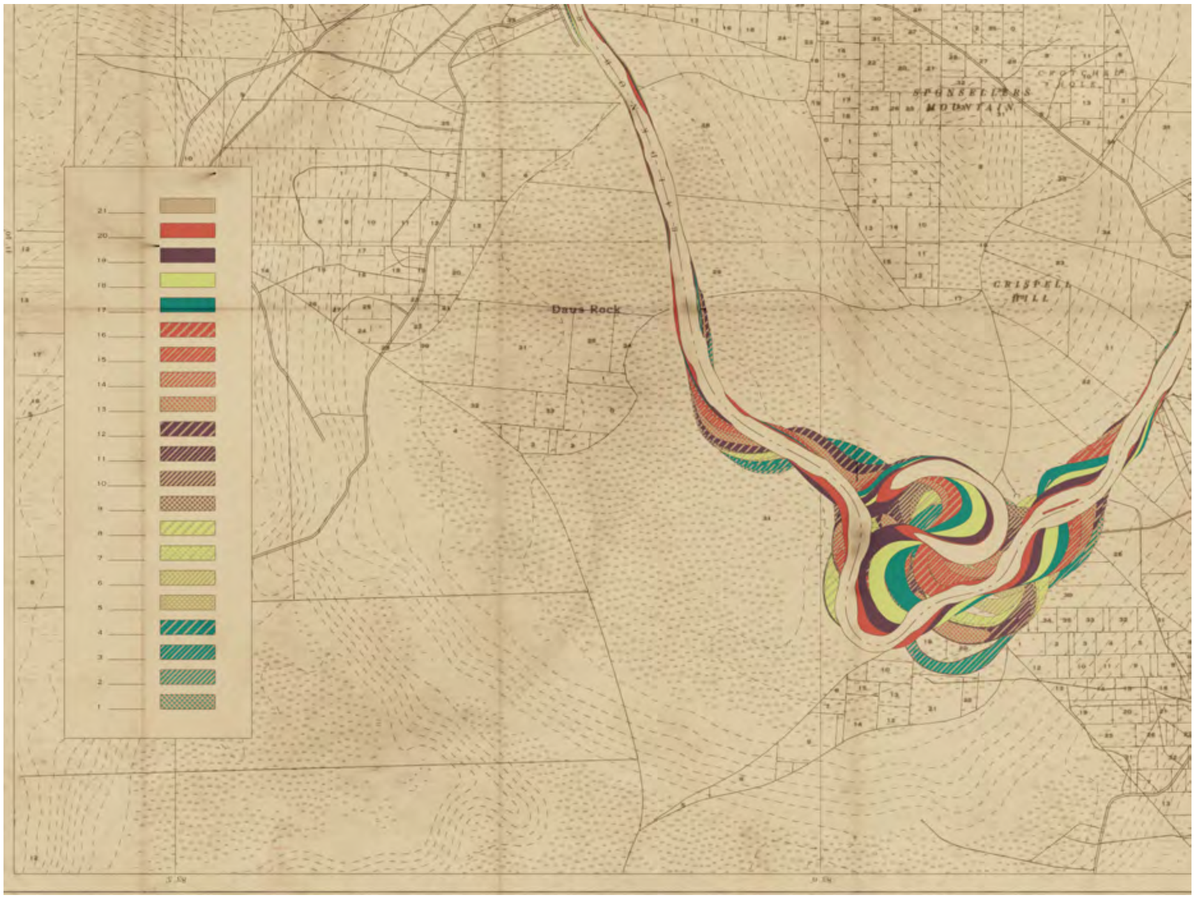


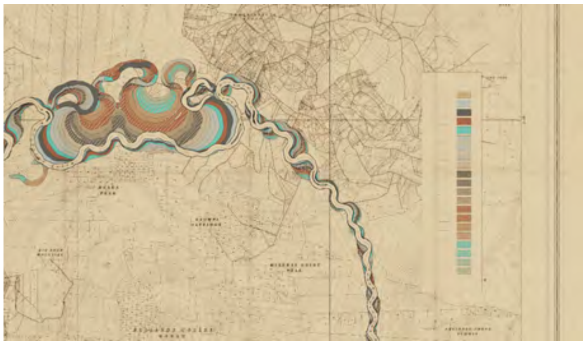
00:16

Robert Hodgin



Robert Hodgin

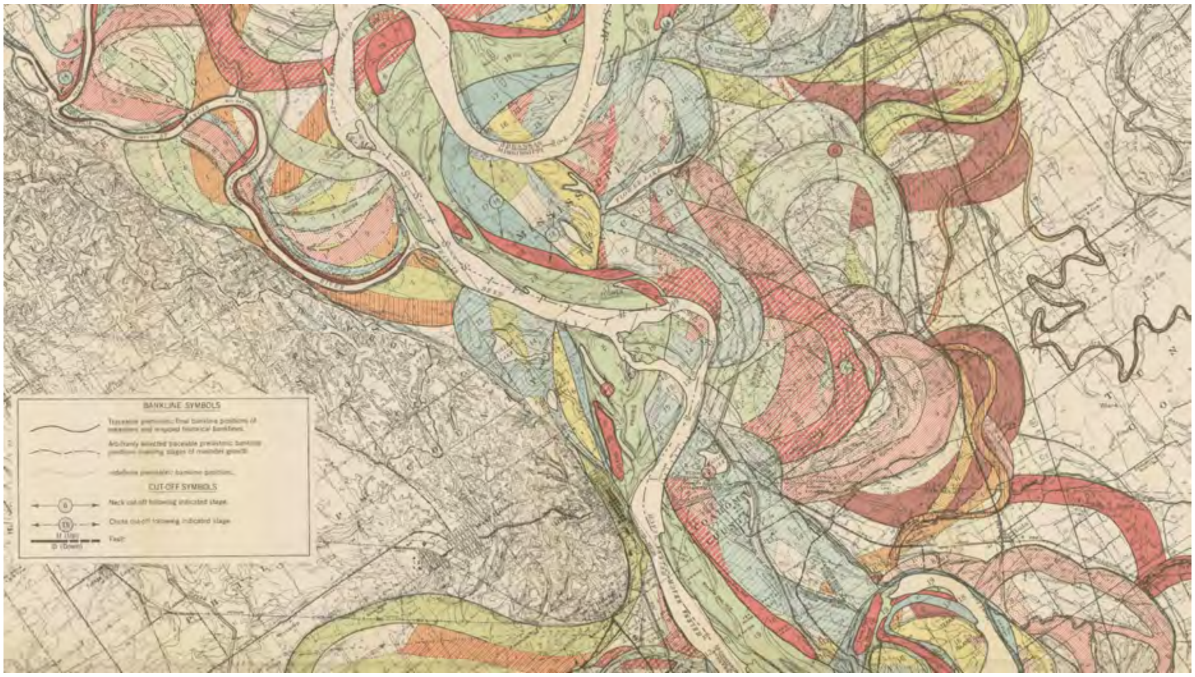





Inspiration

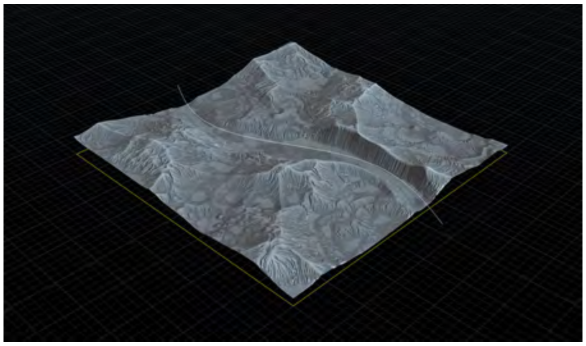
My all-time favorite map-based data visualization was created in 1944. Harold Fisk, working with the US Army Corp. of Engineers, mapped the length of the Mississippi River. What sets his visualization apart from others is that he maps the river through *time*, and

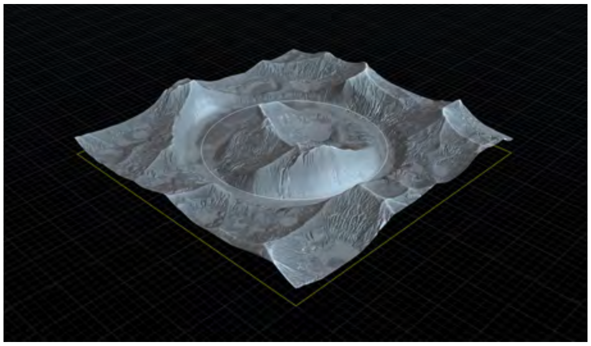
The Alluvial Valley of the Lower Mississippi River (pl. 22, sh. 6, detail), Harold Fisk, 1944


BANKLINE SYMBOLS
Traceable prehistoric final bankline positions of meanders and mapped historical banklines.
Arbitrarily selected traceable prehistoric bankline positions marking stages of meander growth.
Indefinite prehistoric bankline positions.
CUT-OFF SYMBOLS
Neck cut-off following indicated stage.
Chute cut-off following indicated stage.
U (Up)
D (Down)
Fault

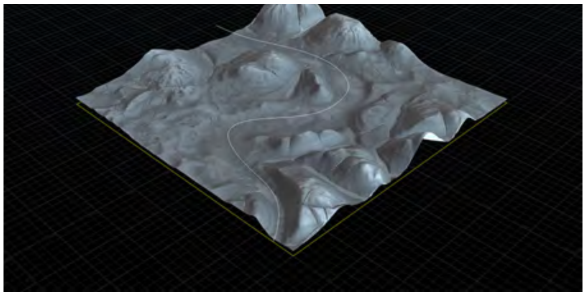
It begins with a line.

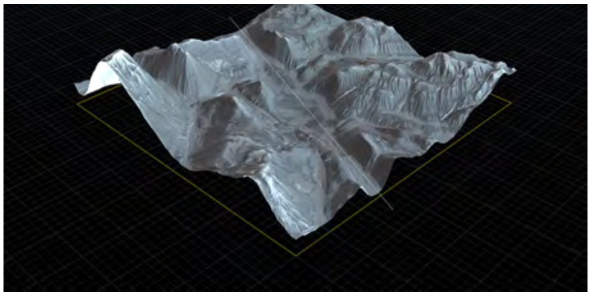
This map generator system was created with Houdini. All the system requires is an input guide line. The line becomes the river and the river moves and evolves over time and leaves behind a visual history of its growth. The map is of a place, albeit a fictional place, so there needs to be terrain on which the river flows. I generate a plausible randomized terrain and use the guide line to cut away a valley. The valley controls the flow of the river: slow rivers on flat plains meander, fast rivers on steep terrain do not.





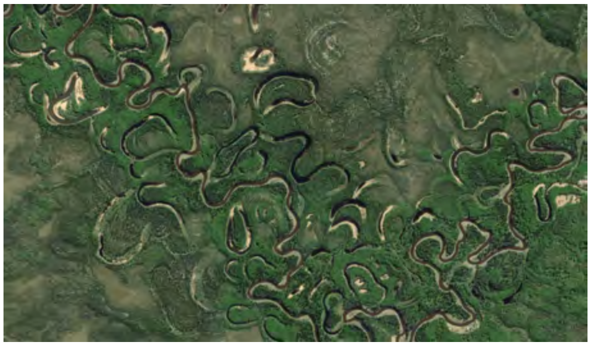
Robert Hodgin





Meandering

As a river moves through the landscape, it is constantly eroding material from the river bed and depositing it downstream. This process continually reshapes the flow of the river causing it to create snaking bends. Browsing through satellite images of the Earth, it is easy to find examples of river meandering. Below are a few examples that will inform our attempts to recreate the geological process with custom code.



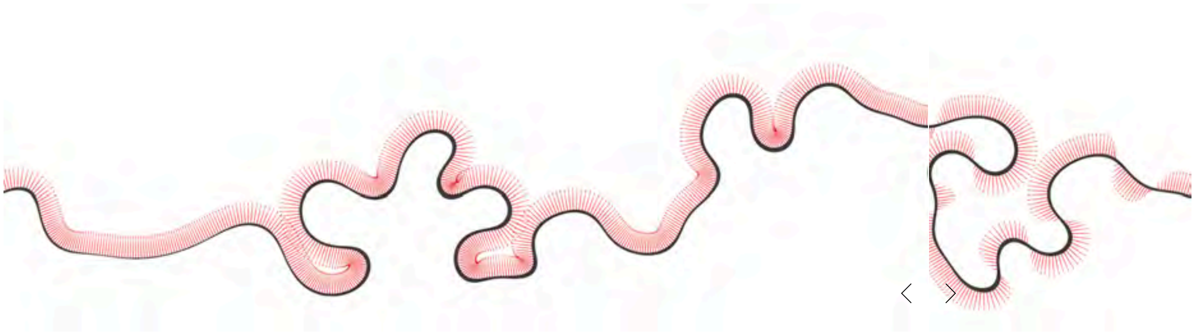
The underlying dynamics of this system are easy enough to understand: rivers pick up material and deposit it downstream, rivers erode the outer bank of a curve where the water speed is higher and deposit on the inner bank where the water speed is slower, and rivers can cut off meanders in favor of a straighter path which forms isolated sections called oxbow lakes.

To simulate this process with code, we can ignore erosion and deposition and just think of a curved line on a plane. Each point on the curve has a coordinate frame which is represented by three vectors: normal, tangent, and bitangent. Since we are dealing with a planar phenominon (ignoring changes in elevation for now), we can disregard the normal vector (in our case, the normal just points straight up) and just make use of the tangent and bitangent vectors. The tangent points in the direction of the line and the bitangent is at a right angle to the tangent.

One thing to note, the bitangent vector isn't enough to make our curve meander. A derived bitangent generally points to the same side of the line, whereas we need our bitangent to be aware of which way the line is curved. It should always point to the outside of the

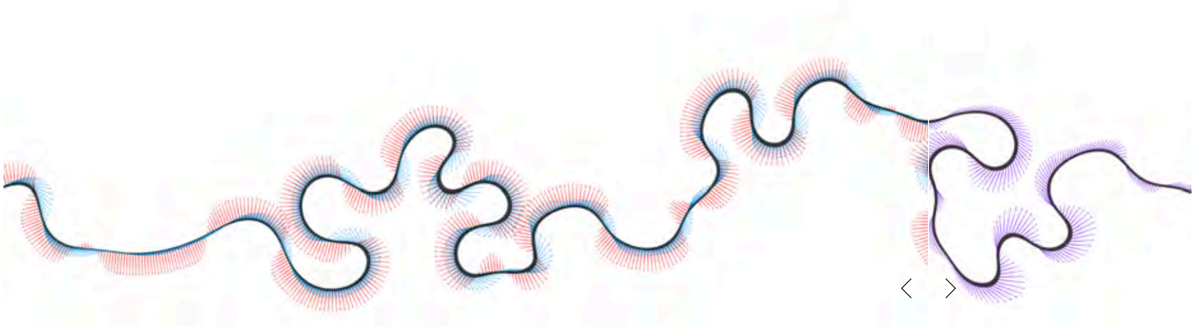
Robert Hodgin

Standard bitangent on the left, modified bitangent on the right

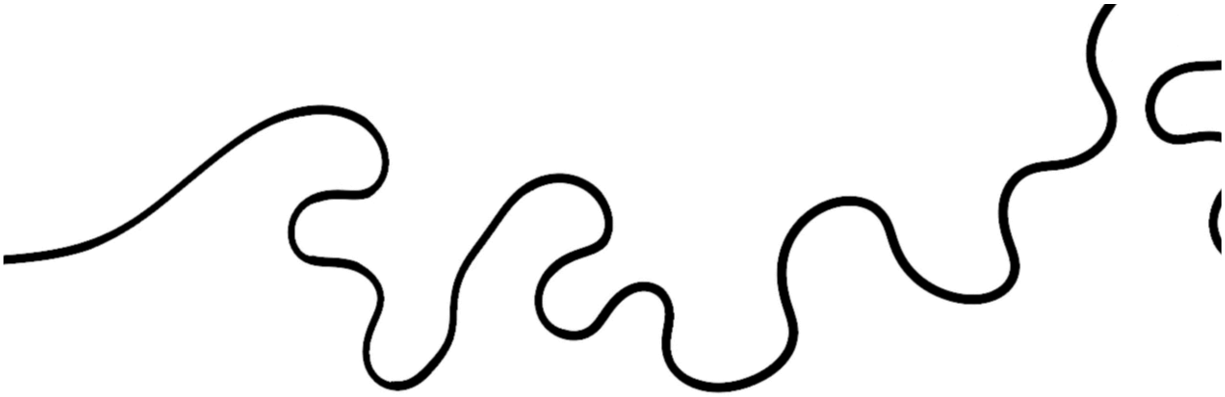


Using the tangent and modified bitangent, we create a new vector that is a blend of the two. This new vector is added to the position of each point on the curve. With this basic logic, the bends in the river form organically. The style of the bends can be influenced by adjusting the overall influence of these two vectors individually, and the intensity of the bends can be adjusted by increasing the scale of the final blended vector.

Tangents (blue) and modified bitangents (red) overlapped on left, new blended directional vector (purple) on right



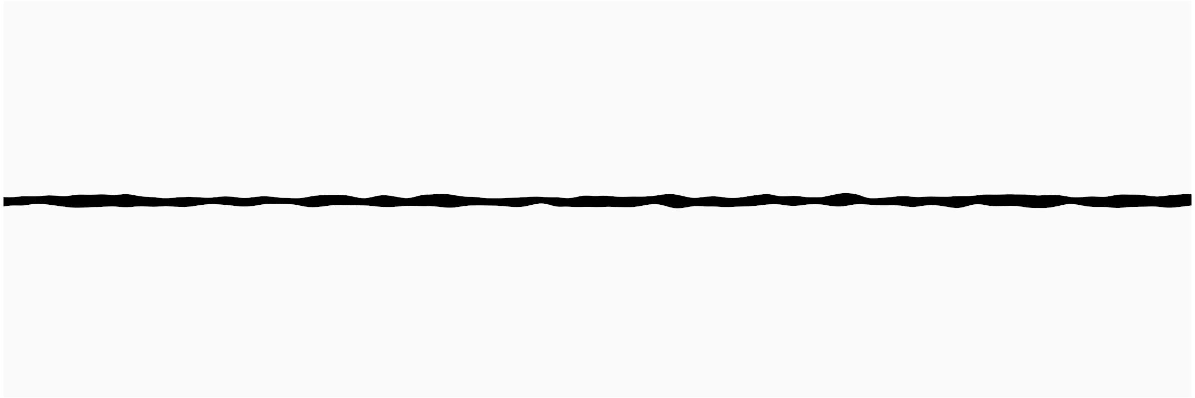
Robert Hodgin



Oxbow Lakes

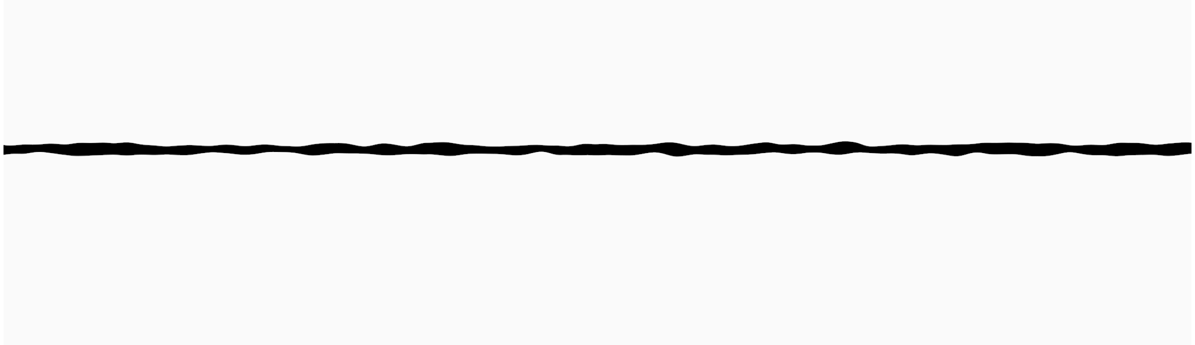
When a river bends back into itself, the meander bend gets cut off in favor of a shorter path. This process creates oxbow lakes which are generally crescent shaped bodies of water to the sides of the main river flow. To find potential oxbows in our river, we do distance checks to see if the river is in danger of overlapping itself. If a near collision is found, the part of the curve between the two collision points is isolated and turned into its own curve segment. The only behavior that these newly generated lakes exhibit is that they get smaller over time until they are deleted from the system. *(The current system doesn't allow for the river to reintersect the oxbow lakes, so an extra step is added to ensure that the lake deletes parts of itself that are at risk of being overlapped by the river.)*

Creation of Oxbow Lakes



Generating the historical record

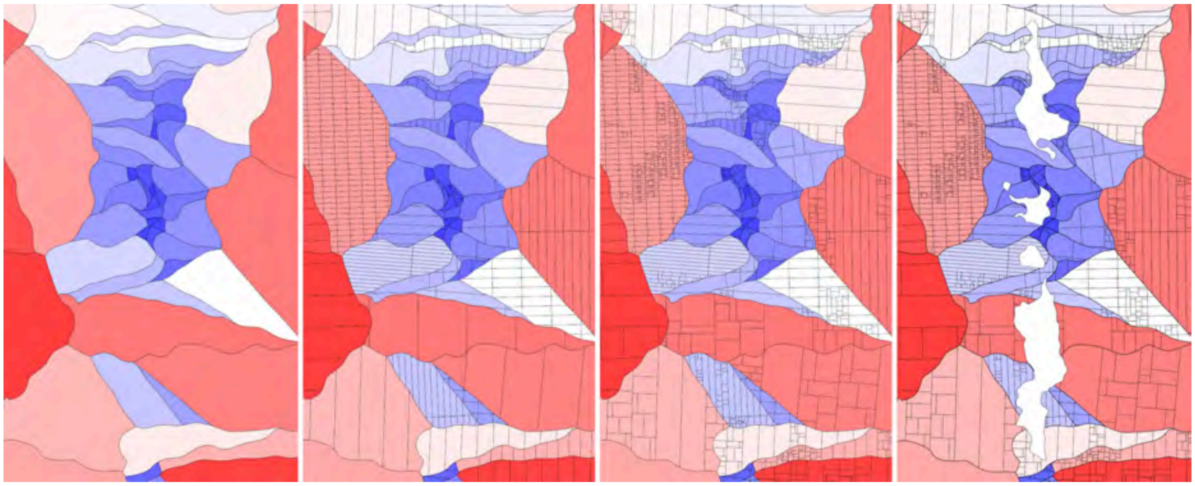
Robert Hodgin



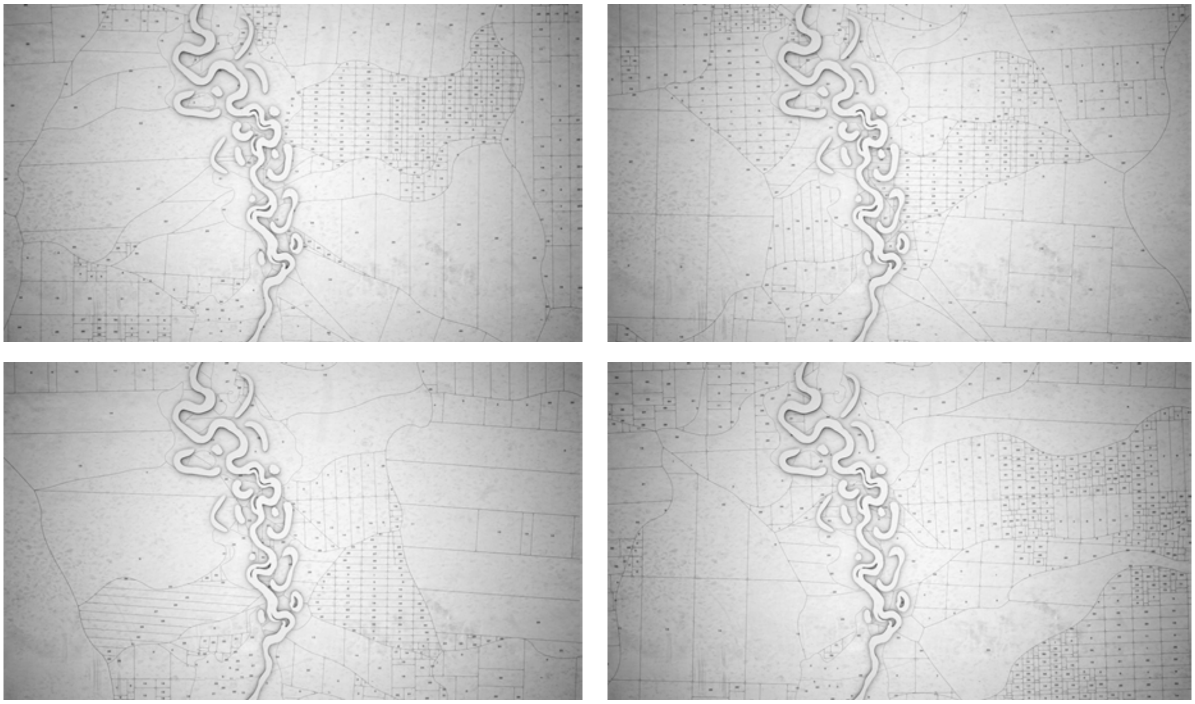
The Land Plots

The background grid-based plots were created by scattering some points on the main river spline and some additional points on the background plane. Voronoi Fracture is used to turn the background plane into smaller polygons with irregular edges. Each polygon is then broken up into a non-uniform grid. Each of those new polygons is broken up further with an iterated subdivision loop which creates plots of pseudo-random sizes. Finally, an organic shape created from the river geometry is used to create boundary plots along the shore. The center point of the final set of polygons is calculated which becomes the location of the plot id number text layer.

The four steps of the land plot generator



Four different land plots by varying a seed variable



Early prototype

Robert Hodgin

The Roads

The network of roads didn't begin as a network of roads. The initial algorithm was an attempt to create an organic growth pattern. The system begins with some randomly placed points. Each point is given a directional growth vector. The point moves along this growth vector and draws a line as it goes. If the point encounters another line, it stops. As the point moves, the directional vector rotates slowly. I found that if I mix curving vectors with non-curving vectors, the result ended up resembling intersecting roads. I emphasized this result by making the thickness of the lines be directly proportional its length.

Robert Hodgin

flight404

00:21

Naming the Geographic Features

The first map images that I created lacked a sense of place. There were no names for any of the places which seemed like a missed opportunity to do something compelling. I considered randomly generated nonsense strings, or manually naming places, but neither of these options seemed like a good scalable solution.

I discussed the problem with Andrew Bell and he found a USGS database (GNIS data) of geographic place names. He then wrote a python script to parse the content and crop off the suffix (things like "Lake" or "Summit") to produce a list of proper names to use for my features. I did not want to name features after places that already exist so cropping off the suffix helps to reduce the likelihood of recreating something called "Hudson River". Instead, new suffixes were randomly added to produce feature names like "Hudson Summit" or "Hudson Swamp". Additionally, he pulled out duplicate entries as well as excessively long names (Little West Fork West Branch Feather River showed up in one of the generated maps).

The features that were isolated for naming include the main river, lakes, islands, peaks, basins, and marshland. The river, lakes, and islands were easy enough since those features are present from the creation of the river system itself. The peaks and basins were isolated by analyzing the slope of the terrain. Flat sections were found and if those flats were near the water level, it was considered a basin. If the flat was at a high elevation, it was designated as a peak. The marshland was slightly more challenging. Areas with low slope were isolated and turned into axis-aligned bounding boxes whose center became the location for the label. Future iterations of this project will find ways to lay the text out in a more map-like manner with text flowing along features like ridges and thin lakes.

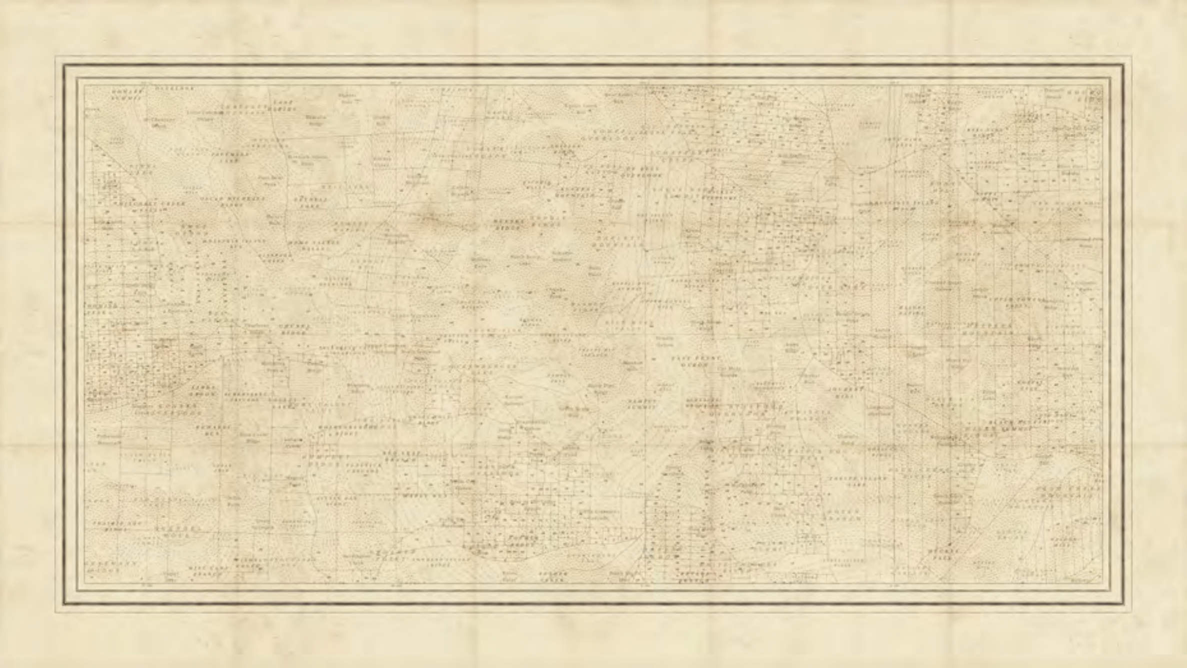
Random scattering of randomized place names to show the variety of results.

Robert Hodgin

North Cody
Oxbow
RUN
HIGER
Pretty Hip
Overlook
SOEHLS
RIDGE
KERRINGTON
TAIL
Black
Gray Eaglenail
Oxbow
FENDER
CREEK

LINKE
OXBOW
SCHERZBERG
CASCADE
VOGLEY
LAKE
HOME VALLEY
FALLS
Sharping
Point
SANDY RICHARDS
OVERLOOK
POND
TWO NATION
CRAZY HOLE
RIDGE
REMALIA
RUN
WOLFENBERGER
RIDGE
Blue Crane
Ridge
Lubeck
Creek
HUMPERT
RIDGE
FLASNICK
CASCADE
RED LEAF
BRANCH
Mazour
Pond
North Cody
Oxbow
Dillie
Falls
LITTLE DAD
RIDGE
MEELY RUN
Stively
Overlook
Pretty Hip
Overlook
DEL

SANDY RICHARDS
OVERLOOK
POND
Kucera
Summit
Black Pipe
Ridge
DEMP
SUMM
South Scalp
Run
TWO NATION
CRAZY HOLE
RIDGE
Braunkweiger
Oxbow
Dillie
Ridge
CASCADE
Lubeck
Pond
RED LEAF
BRANCH
Pishel
Pond
ASNICK
SCADE
MANNOX
OVERLOOK
North Cody
Oxbow
MEELY RUN
Bear-in-the-Lodge
Rapids
Little Coleman
Cascade
Pretty Hip
Overlook
SOEHLS
RIDGE
Gray Eaglenail



Robert Hodgin

There are some features that I would like to implement which would require a rework of the meandering logic itself. The current system is not as robust as I would like it to be. I'm considering moving from a spline-based approach to a particle-based one which might make branching rivers possible as well as natural lake formation. The particle based approach could also make river deltas possible as well as allowing rivers to re-intersect existing oxbows.

I would also like to work on making the road system an iterative approach. The first pass would lay out the major highways, the second pass would build secondary roads off the first pass, etc. I also need to work on the logic for how roads should avoid growing up steep slopes.

PREV
Akari Lanterns

NEXT
Traffic