Redistricting using Active Contours
CS269 - Term Project

Prateek Malhotra Mihir Mathur Zeina Migeed Alexandre Tiard

December 2019

Abstract - The problem of assigning district boundaries to ensure fairness and legality of boundaries is a very important problem, especially in the upcoming 2020 United States redistricting cycle. In this project, we examine existing computational redistricting methods and propose a new method that uses Chan-Vese active contours and K-Means clustering for drawing district boundaries. As compared to previous approaches, our method introduces more flexibility by redistricting based on a scoring function which can be specified by the user.

1 Introduction

1.1 Problem setting

Gerrymandering is the process of redrawing district boundaries with a mind for political gain, by skewing the transformation of votes to seats. This phenomenon, which poses a democratic problem by heavily influencing election results on the state level, has seen rising public interest over the past 20 years in the United-States. This culminated with a case that reached the Supreme Court: *Gill v. Whitford*, which poses a question: are traditional methods of redrawing district lines fair?

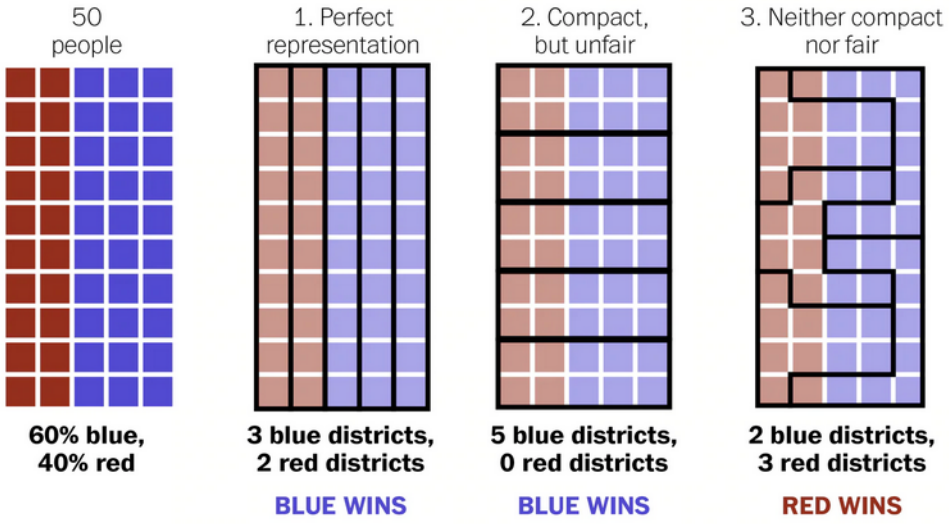


Figure 1: An example of gerrymandering

Implicitly, this assumes that it is possible to distinguish a state legislature's efforts to apply traditional districting criteria from partisan gerrymandering, and that an appropriate metric exists to do so. In fact, this is not as straightforward as it may seem.

1.2 Identifying gerrymandering

There are mainly 2 techniques used when performing partisan gerrymandering: packing and cracking. Packing refers to concentrating the opposition voters into one district, which is only worth one vote, leaving the balance of surrounding districts against their favor. Cracking refers to spreading the opposition party's constituents across several districts to create several minorities, hereby increasing the chance of cancelling their votes. A simple unfairness metric could therefore be asymmetry, as shown in Figure 1. In other words: similar vote shares should result in similar seat shares. However, as noted by several Justices, "asymmetry alone is not a reliable measure of unconstitutional partisanship" [4]. Asymmetry may arise for several reasons that aren't linked to partisan manipulation, including the application of traditional criteria. The geography of the parties' supporters is a nonpartisan cause of asymmetry [15], as would be an attempt of carving out districts in which minority groups can elect chosen candidates [3].

Another issue is that, to a party that is undertaking unconstitutional district manipulation, an inherently subjective risk evaluation is required. The highest potential gain in a bipartisan system is obtained when the opposing party's voters are spread out in every district so as to be just below 50%. This is also a very high risk scenario: a small fluctuation in voter patterns might overturn the result in all districts. Therefore, there is no single optimal way to gerrymander, which makes the practice harder to recognize.

1.3 Computational redistricting as a solution

There have been calls to use an algorithm to redraw congressional lines as a way to avoid this problem. If such an algorithm was to be written, and if the nation agreed to use it, there should be no need to recognize gerrymandering. There are, in the literature, mainly 5 arguments in favor of computational redistricting. First, it is argued in [9] that computational redistricting creates a neutral and unbiased district map. Second, computational redistricting prevents manipulation by removing political actors from the process of choosing district plans, while simultaneously producing districts that meet specified social goals . Third, computational redistricting promotes fair outcomes by pushing political debate to be over the general goals of redistricting, rather than over particular plans - where partisan interests are most likely to be manifest. Fourth, computational models provide a recognizably fair process of meeting any representational goals that are chosen by the political process. Finally, computational redistricting eases judicial and public review by promoting transparency; and because automation processes creates a clear separation between the intent and effect of redistricting.

However, there have been some push-backs and arguments against this seemingly perfect solution, that centered around 2 points. In Dixon v United-States (1968), it was argued automated processes, even if based on nonpolitical criteria, may have politically significant results. After all, redistricting is inherently political, and automating the process will not change that fact. This is not an argument against the use of automatic methods for redistricting, but rather an argument that aims at setting realistic expectations - they can be used to reach a neutral goal effectively, nothing more. The second argument was that political bias will never be removed from the legislative process, since the action of choosing an automatically generated plan will be political.

From a purely practical standpoint, another problem arises: algorithmic complexity. Even when using simplified definitions of *fair* and *legal*, so as to circumvent the aforementioned problems, showing that a given districting plan is fair among legal maps is NP-hard [8]. This holds in spite of reducing the list of criteria to districts that are loosely compact with equal population. To see where the problem lies, one can naively cast the problem as a search of n population blocks to draw r districts. The goal is therefore to find :

$S(n,r) = \frac{1}{r!}\sum_{i=0}^{r}(-1)^i(\frac{r!}{(r-i)!i!})(r-i)^n$.

1.4 A fair computational model

Recent efforts have been made to propose a computational redistricting method that would take into account both fairness and legality. Generally, the criteria taken into account are compactness and equal population, as these are the only objective metrics. These methods fail to address the concerns expressed in the previous paragraphs by solely producing districts that are *compact* - according to an arbitrary metric - and of equal population. They notably fail to take into account statistical details of legal importance, such as the preservation of historical districts and representation of minorities. We argue that this makes their method irrelevant: since they cannot be used by the legislative branch, another solution has to be found.

What we propose is to create a model for computer-aided redistricting, that provides an efficient solution to the problem of creating districts of equal population. This model should be flexible so as to allow users to specify racial and historical constraints, thereby making it usable. This model should have a high level of randomness, so as to keep focus on the goals of redistricting, as predictability will push the debate towards the quality of specific plans. This model should be transparent, so as to make the users' constraints and their effect clear.

We cast this problem as a computer vision task, and use a Chan-Vese formulation of active contours, introduced in [2]. We show that this method can produce a set of viable district lines, and argue that this serves as a proof of concept of a computer-aided redistricting system.

2 Related Work

Computational methods have emerged as a solution to unconstitutional partisan redistricting, and are based on clustering algorithms. These algorithms have a goal of producing clusters of equal population that are maximally compact, which are necessary but not sufficient conditions. They mostly vary in the way that they corner edge cases and interpret legal issues, such as representation of minorities. All of these methods rely on data provided by the Census Bureau, which we will present now.

2.1 The Census Data

The baseline data used for those tasks is provided by the American Community Survey (ACS), which is an ongoing survey that covers a broad range of topics about social, economic, demographic, and housing characteristics of the U.S. population. The data is agglomerated over 5 year periods, which provides statistical robustness for smaller geographies.

The dataset is stratified in 87 different levels and contains 578,000 geographic areas. Such levels include nation, all states (including DC and Puerto Rico), all metropolitan areas, down to tracts and block groups. Each geography is labelled with a GEOID, which allows to project the data back to a map. Detailed tables, containing 20,000 variables, notably record population counts at every level. //

2.2 Performance Metrics

There are metrics that are required to assess performance: compactness and population distribution. While verifying that the districts produced respect the equal population criterion, compactness is harder to define, and has been a topic of research of the past decades ([1], [7]). In spite of the existence of a variety of methods to compute it, compactness is defined as a score ranging from 0 to 1 in

which 1 generally represents a more compact district. This score is obtained by taking the ratio of the district area to the area of some convex shape. For example, the first methods were to use the ratio of the area of the given district to the area enclosed in its convex hull, or of its minimum bounding rectangle. Two metrics, which use the same principle, have become baselines. The first was proposed in [14], where the author uses the ratio of a district's area to the area of its minimum bounding circle - which defines the Reock measure. While this idea is clear, minimum bounding circles are hard to compute. Thankfully, an efficient algorithm was proposed in [5]. The second was proposed later, in ([13], authors define the Polsby-Popper (PP) score as the ratio of the area of the district to the area of a circle whose diameter is the perimeter of the district. We will, arbitrarily, use the Reock measure in this paper.

2.3 Clustering methods

The first step when performing redistricting via clustering is to process the maps provided by the Census Bureau, transforming them into point clouds, where each point represents a block group and hereby a given population. The point is placed at the barycenter of the polygons representing the block groups, and to it is assigned a weight which corresponds to the represented population.

In [6], the authors use a weighted version of K-means++ to initialize a number of centroid corresponding to the number of existing districts, and then run the algorithm on the point cloud of the corresponding state. K-means does optimize for equal cluster partitions, and therefore the authors add a layer to the algorithm, which scales clusters based on their represented population. In doing so, they add two hyperparameters, to which results are sensitive. Another drawback of their method is the way in which they validate their results: they simply compare the ratio of pairwise distances within districts between their solution and the current districts. As discussed in the introduction, this only offers a partial solution to the problem. In particular, they ignore all problems related to the representation of minorities and the preservation of historical districts, which makes their method hard to use in practice.

In [3], authors instead use hierarchical agglomerative clustering to create districts, with a cut at the desired number of districts. This seems like a more natural choice: this algorithm optimizes a cost which directly measures cluster compactness. It does not guarantee that the overall population will be split uniformly amongst different clusters however. To resolve this issue, the authors propose to reassign points to neighboring clusters to balance out the result, and to do so choose the points that are closest to their cluster border - which are the points that will minimally impact compactness.

To analyze their results, and compare them to the district lines that currently exist, they simulate 1000 cuts through their method, and produce statistics of the resulting Reock scores. This allows them to measure where on the distribution the current districts lie. Tails of the distributions are a stronger indicator of gerrymandering than simple symmetry. They then project voter data into their redrawn maps, and plot the distribution of symmetry between the number of votes and number of seats, as shown in Figure 5. While this is a very interesting way to perform comparison, the authors don't find a way to handle historical districts, or minority representation. Instead, they choose to manually fix these districts, without measuring the effect that these constraints have on the result. In practice, using this method requires a lot of human input and prior knowledge: the user would have to specify which districts to preserve for every state, which is a non-trivial, political problem.

We therefore advocate for an algorithm that would help lawmakers in redrawing district maps. This algorithm should make the constraints and assumptions that lawmakers work under perfectly clear, and measure the effect of those constraints. We present a proof of concept of this method in the next section.

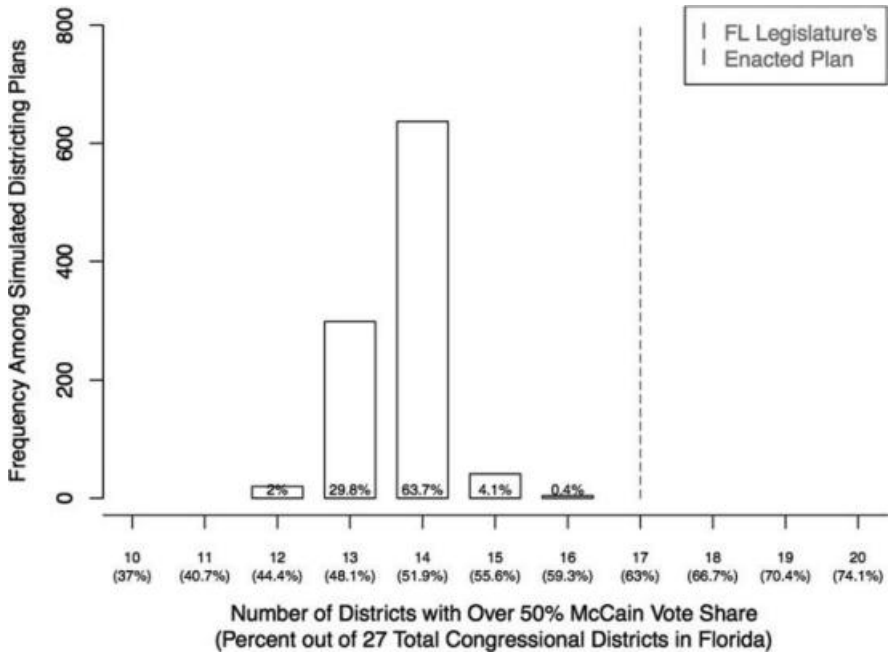


Figure 2: Results of 1,000 simulated districting plans

3 Our Approach

3.1 Constructing Cartesian Image

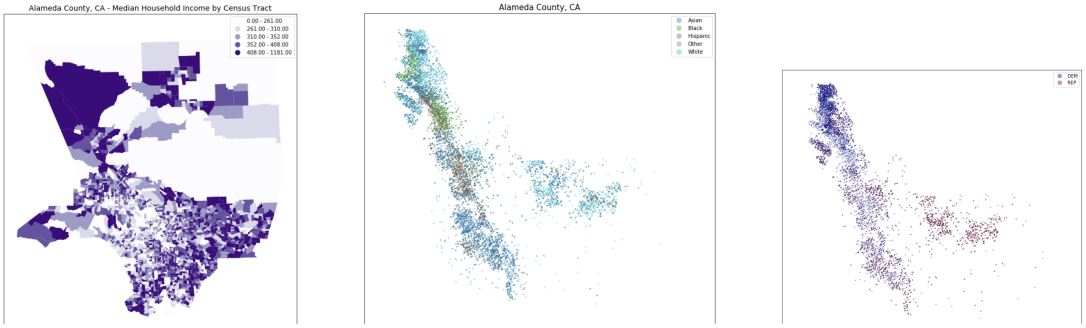
In this section we discuss the map generation process then we will show how to redistrict a given map using active contours.

The first is to generate a map of a county. For this, We used TIGER data to map from counties (GEOIDs) to polygons and then projected these polygons to construct a map of the county. The next step is to project the population onto the map. We then used ACS-5 data to generate data about people to project it on the previously created map. ACS-5 has 18,000 variables regarding people's orientation, communication patterns, movement, and beliefs.

This gives us access to a large amount of data which can be used to make decisions or grant district scores. For example, demographic data related to a population's racial composition can be used to prevent racial gerrymandering [10] and median income based maps can be used by an unbiased reviewer to understand signs of polarization [11]. Below are some examples of the type of data we can generate. All the code for generating the maps can be found on: https://github.com/Mihirmathur/redistricting/tree/master/viz_population

3.2 Finding District Centers

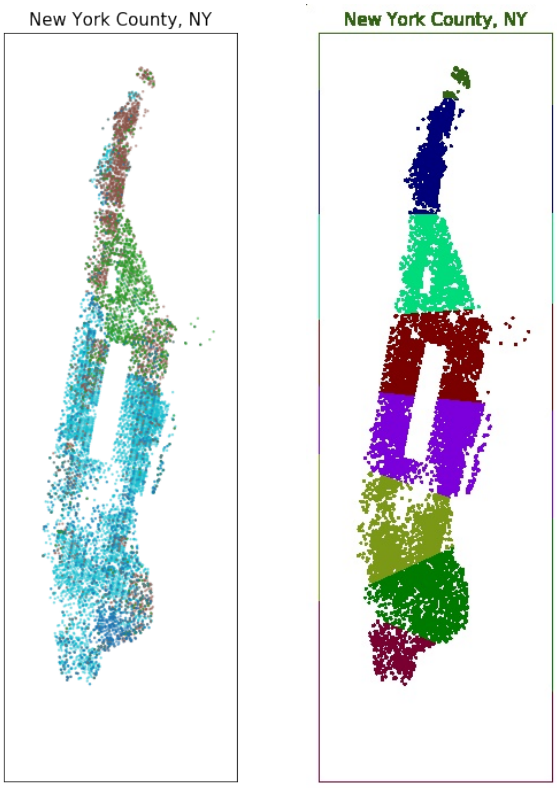
After generating a map, the first step is to find n centers to partition the map into n districts. One approach could be to allow the user to specify the centers manually. However the user may specify centers such that it is impossible to redistrict the map into n districts, such as if the centers are concentrated on a particular area of the map. Additionally, the user could try to pick centers in such a way that the map is gerrymandered. Therefore, our approach is to run the K-Means clustering algorithm for finding good candidate centers for districts. We then use these centers for initializing circle level set of ACWE. In the next section, we discuss how to use these centers to create active



((a)) Median income ((b)) Racial and Ethnic ((c)) Voting preference

Figure 3: Projecting census data on maps - Alameda County, CA

contours for redistricting. Here is an example of using K-Means to cluster the population data directly:



((a)) Racial and Ethnic ((b)) clustering with K= 8

Figure 4: KMeans based redistricting on New York City

3.3 Active Contours

The Chan-Vese active contour model is a solution developed for image segmentation, aimed at solving the following problem:

$$inf\{F^{MS}(u,C) = \int_{\Omega}(u-u_0)^2 dxdy + \mu \int_{\Omega - C} |\nabla u|^2 dxdy + v|C|\}$$

where each Ω_i is a connected component of the image u_0, and u is piecewise-smooth within each Ω_i. The restriction of this problem to piecewise-constant functions leads to the minimal partition problem, and the Mumford and Shah functional [12]:

$$E^{MS}(u,C) = \sum_i (u-c_i)^2 dxdy + v|C|$$

We minimize

$$F(c_1,c_2,C) = \int_{\Omega_1=\omega}(u_0(x,y)-c_1)^2 dxdy + \int_{\Omega_2=\Omega-\omega}(u_0(x,y)-c_2)^2 dxdy + v|C|$$

3.4 Utility Function

Our scoring function ensures compactness and assigns a score based on how compact a given contour is. Whole the function currently only focuses on compactness, it can easily be adjusted to include additional criteria. We will discuss our scoring function in this section.

Suppose we want to divide some map m into d districts. Let c_m denote a contour for m and let $|m|$ be the total population and $|c_m|$ be the population within the contour. Our goal is to define a utility function $\mathcal{S}$ which assigns a score $s \in [0,1]$. Intuitively, the score should increase whenever the population of our contour $|c_m|$ approaches the ideal population I. If the $|c_m|$ is much larger than I then the score should decrease again. So we define our function as follows:

$$\mathcal{S}(c_m,d) = \begin{cases} 1 & \text{if } |c_m| = I(m,d) \\ \mathcal{S}(c_m,d) = min\left(\frac{|c_m|}{I(m,d)}, \frac{I(m,d)}{|c_m|}\right) & \text{otherwise} \end{cases}$$

where

$$I(m,d) = \frac{|m|}{d}$$

We can see that in the first equation, the first case checks that the population of the contour is ideal. Otherwise, we calculate the score based on the ratio. The first term of the minimization ensures the population is not too small and the second term ensures the population is not too large. So in our approach, the size of the contour can adjust based on the score.

3.5 Integrating Chan-Vese Active Contours with Utility Function

We need to ensure that our contours do not overlap and we also ensure that the contours cover the entire map. For the first problem, we simply remove the area segmented by previous contours from the image so that it may not be included in any other contours. is demonstrated in Figure 5. To do so, we define the area delimited by the previous active contour as a mask and set all pixels to maximum values in that area. This effectively removes all information content in the area, which will therefore be disregarded for the next iteration of our algorithm.

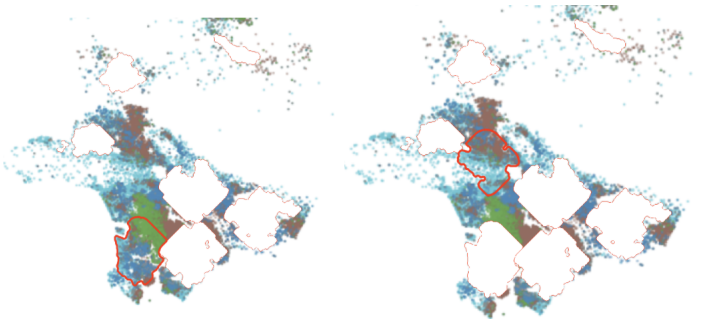
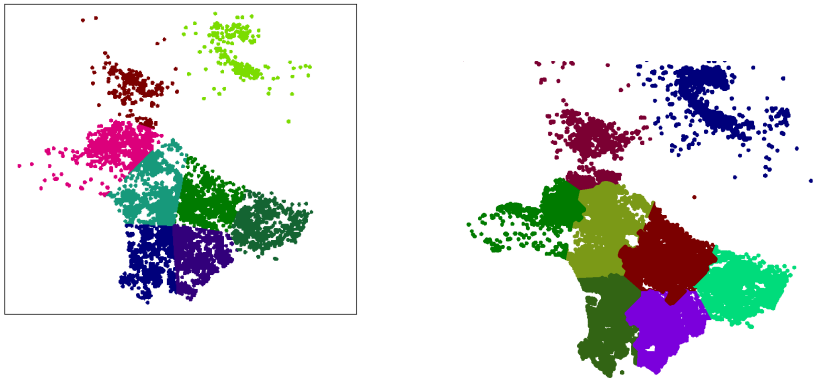


Figure 5: Ensuring non overlapping contours by removing already created districts



((a)) KMeans

((b)) Chan Vese + Agglomerative clustering (Ours)

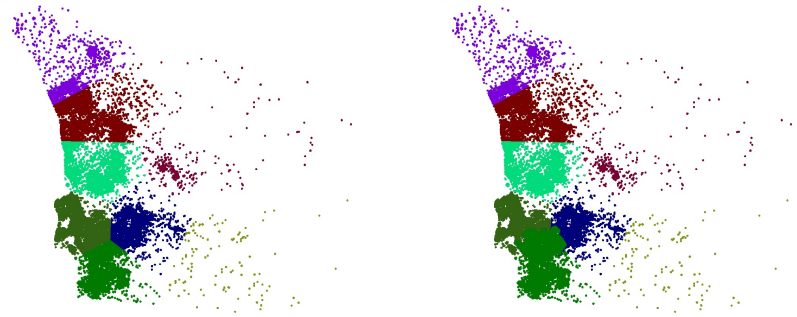
Figure 6: Redistricting for Los Angeles County, CA

3.6 Agglomerative clustering for completeness

The previous method provides no guarantees of assigning all points to a cluster. To complete the contours, and to make sure that all points are assigned, we use agglomerative clustering - using the districts produced by our active contours as a highly variable base. To do so, we maintain a secondary boolean array to mark which nodes have been assigned. After all iterations have completed, this array contains the information of unassigned points. We compute Ward's distance of these points - using a Euclidean norm - to the clusters, and assign them to the closest cluster. This is done to maximize compactness, and ensures that all points have an assignment.

4 Results

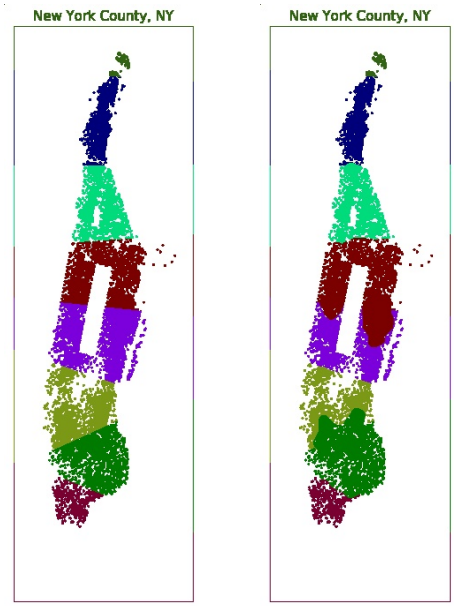
We show our results for redistricting three counties. Los Angeles, San Diego and New York in figures 6, 7 and 8 with comparisons to the k-means method. In the figures, we can see the results of redistricting using Chan Vese and Agglomerative clustering.



((a)) KMeans

((b)) Chan Vese + Agglomerative clustering (Ours)

Figure 7: Redistricting for San Diego County, CA



((a)) KMeans

((b)) Chan Vese + Agglomerative clustering (Ours)

Figure 8: Redistricting for New York City

5 Future work

Our approach can be extended to perform more accurate redistricting. In particular, the utility function can be modified to include additional criteria for redistricting. For example, we can impose additional restrictions on the shape of the contour or include information about the diversity of the population within a given contour in the score.

Defining compactness also proved to be a challenging problem. Our approach assumes that the population is uniform across the colored components of the map. That is that every dot on the map represents a fixed population, but this is not necessarily true. So compactness can be redefined to accurately compute the population in a given area, and this can easily be integrated with our utility

function. We leave this for future work.

Finally, we propose a way to measure fairness for future work. First, we randomly generate a set of partitions over a given map and project the voting data onto the map, generating a Gaussian distribution where the tails denote the outliers. We now consider our current partitioning as a result of active contours and check where the partitioning lies within the distribution. Intuitively, if the partitioning lies on either ends then the partitioning is an outlier so the redistricting is biased. The goal would be to produce a partitioning that lies in the center.

6 Conclusion

In this paper we pose the problem of redistricting to avoid gerrymandering as a computer vision problem and rely on Chan-Vase Active Contours for redistricting. Our results show that active contours produce compact districts while preventing redistricting using irregular shapes as shown in example 3 of Figure 1. Furthermore, our method is more efficient and less susceptible to gerrymandering compared to manual redistricting. We leave the problem of formalizing and measuring fairness of redistricting to future work.

References

[1] Ronald R Boyce and William AV Clark. The concept of shape in geography. *Geographical Review*, 54(4):561–572, 1964.

[2] Tony F Chan and Luminita A Vese. Active contours without edges. *IEEE Transactions on image processing*, 10(2):266–277, 2001.

[3] Jowei Chen and Jonathan Rodden. Cutting through the thicket: Redistricting simulations and the detection of partisan gerrymanders. *Election Law Journal*, 14(4):331–345, 2015.

[4] Carmen Cirincione, Thomas A Darling, and Timothy G O'Rourke. Assessing south carolina's 1990s congressional districting. *Political Geography*, 19(2):189–211, 2000.

[5] Bernd Gärtner. Fast and robust smallest enclosing balls. In *European Symposium on Algorithms*, pages 325–338. Springer, 1999.

[6] Olivia Guest, Frank J Kanayet, and Bradley C Love. Gerrymandering and computational redistricting. *Journal of Computational Social Science*, 2(2):119–131, 2019.

[7] Aaron Kaufman, Gary King, and Mayya Komisarchik. How to measure legislative district compactness if you only know it when you see it. *American Journal of Political Science*, 2017.

[8] Richard Kueng, Dustin G Mixon, and Soledad Villar. Fair redistricting is hard. *Theoretical Computer Science*, 2019.

[9] Franklin F Kuo and James F Kaiser. *System analysis by digital computer*. Wiley, 1966.

[10] David Lublin. *The paradox of representation: Racial gerrymandering and minority interests in Congress*. Princeton University Press, 1999.

[11] Nolan McCarty, Keith T Poole, and Howard Rosenthal. Does gerrymandering cause polarization? *American Journal of Political Science*, 53(3):666–680, 2009.

[12] David Mumford and Jayant Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on pure and applied mathematics*, 42(5):577–685, 1989.

[13] Daniel D Polsby and Robert D Popper. The third criterion: Compactness as a procedural safeguard against partisan gerrymandering. *Yale L. & Pol'y Rev.*, 9:301, 1991.

[14] Ernest C Reock. A note: Measuring compactness as a requirement of legislative apportionment. *Midwest Journal of Political Science*, 5(1):70–74, 1961.

[15] Jonathan Rodden. Geography and gridlock in the united states. *Solutions to Political Polarization in America*, pages 104–20, 2015.