Characterizing Wildfire Perimeter Polygons from QUIC-Fire*

Li Tan[1], Raymond A. de Callafon[1], and Ilkay Altıntaş[2]

[1] Dept. of Mechanical and Aerospace Engineering
University of California San Diego, La Jolla, CA, U.S.A.
{ltan,callafon}@eng.ucsd.edu
[2] San Diego Supercomputer Center
University of California San Diego, La Jolla, CA, U.S.A.
{altintas}@ucsd.edu

Abstract. QUIC-Fire is a modern fire simulation tool that can simulate the progression of three-dimensional fuel consumption over a landscape, modeling the interaction of a wildfire with weather such as wind conditions around the wildfire. The resulting simulation gives a detailed progression of the consumed three-dimensional fuel that can be eloquently mapped to an image of a burn area in the landscape as the wildfire progresses over time. Although an image of burned vegetation over a landscape gives detailed information of the activity and coverage area of a wildfire, a numerical characterization of the boundary of the burn area can be used for a variety of computations. The boundary of the burn area, also labeled as the wildfire perimeter, can be parametrized with a closed polygon. The set of ordered vertices of the closed polygon provide a compact numerical representation of the location of the wildfire and can be used for computations related to fire coverage area and modern wildfire assimilation techniques to improve the prediction of wildfire progression. Designing a robust algorithm to create a wildfire perimeter in the form of a set of ordered vertices of a closed polygon around the image of consumed vegetation in a landscape is not a trivial task. This paper discusses the properties of two such algorithms: the iterative minimum distance algorithm (IMDA) and quadriculation algorithm (QA) to obtain a closed polygon for a wildfire perimeter. To illustrate the effectiveness, these two algorithms are applied to multiple image (raster) data of a burn area in the landscape of a wildfire created by QUIC-Fire simulations. It is shown that both algorithms are robust in computing wildfire perimeters, and computational time are less than one second for each image created by QUIC-Fire. As such, this work contributes to the development of computational methods to automate the process of characterizing the closed polygon of a wildfire perimeter based on burn area images.

Keywords: Wildland Fire · QUIC-Fire · Polygons · Automation.

* Work is supported by WIFIRE Commons and funded by NSF 2040676 and NSF 2134904 under the Convergence Accelerator program.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_44

1 Introduction

Vegetation dispersed over a landscape is the main fuel component that drives many wildfires. As a wildfire consumes this fuel under the influence of external wind and other weather conditions, it creates a 'burn area' or 'burn scar' of consumed fuel in the landscape that can cause significant damage, economic loss and environmental impacts. Clearly, understanding the wildland fire behavior and reducing the effects of wildfires by either controlling vegetation via prescribed burns or improving predicting the progression of a wildland fire are desirable.

Improving the prediction of wildfire progression has been an active area of research [7,11,12]. Data assimilation by combining wildfire modeling and ensemble Kalman filter is applied in [4, 15, 17], while several studies on the influence of wind condition and fuel have been conducted [2, 18, 19]. Many fire behavior models have been developed to improve the prediction of the wildfire progression [1, 5, 9, 10, 14], and the focus on controlling wildfires by prescribed burns is driven by QUIC-Fire [10]. QUIC-Fire can serve as a modern fire simulation tool to simulate the progression of three-dimensional (3D) fuel consumption over a landscape, while also approximating the dynamic interaction of a fire with weather including wind conditions in the atmosphere around the fire. QUIC-Fire can also takes into account the interactions between multiple fires and can compute fire progression at the resolution of one meter.

A wildfire perimeter, defined as a closed polygon around the burn area of a wildfire or prescribed burn, is an important numerical characterization of the impact of the fire and can be used for a variety of computations. Most wildfire perimeters are obtained from 2D images [3, 13, 21], while the consumed 3D fuel created by QUIC-Fire simulation is mapped to a 2D image of a burn area in the landscape as the wildfire progresses over time. So even for the output of QUIC-Fire, it is desirable to create an algorithm to compute the closed polygon of the wildfire perimeter.

Designing a robust algorithm to create a wildfire perimeter in the form of a set of ordered vertices of a closed polygon around the image of consumed vegetation in a landscape is not a trivial task. Edge detection methods have been applied to wildfire images [16, 20], but only find a set of unordered boundary points that is not suitable to produce a closed polygon. In addition, a wildfire perimeter may include one main closed polygon and multiple additional closed polygons due to sporadic fire spread caused by embers and no assumption can be made on the shapes of the polygons. Due to this complexity of multiple wildfire perimeters, traditional pattern recognition algorithms [6, 8] are not directly applicable.

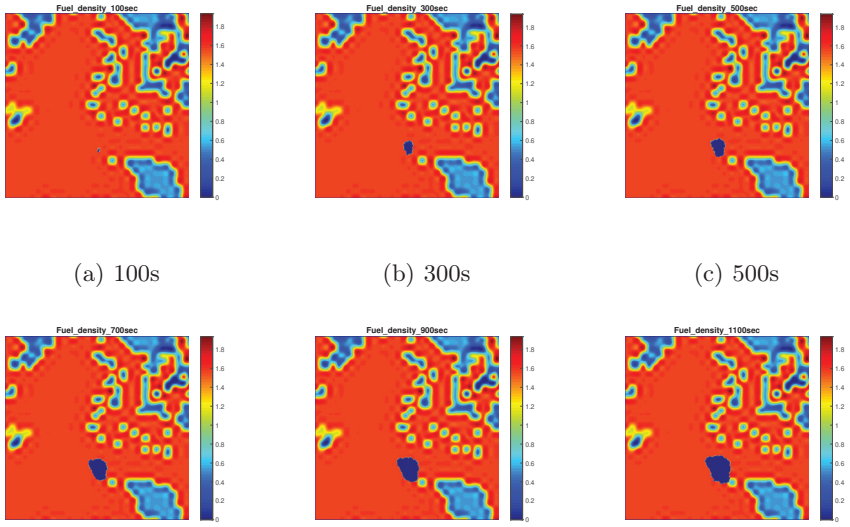
This paper discusses the properties of two algorithms: the iterative minimum distance algorithm (IMDA) and quadriculation algorithm (QA) to create a closed polygon of a wildfire perimeter. The IMDA is based on continually connecting two closest points in the set of unordered boundary points determined by conventional image edge detection. A threshold value is set up to assist in determining whether two points in a cluster are closely located. If one cluster is far away from other clusters, then it is regarded as a new isolated polygon rep-

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_44

resenting a separate fire perimeter due to embers. From a completely different point of view, the QA creates a polygon by recursively dividing the raster image into indivisible rectangles, where all the internal pixels of the rectangles have the same color, and then merging adjacent rectangles that have the same color. In general, QA avoids the process of ordering the unordered boundary points, but takes a longer time when merging the different polygons.

2 QUIC-Fire Output Data

As mentioned in the introduction, the focus of this paper is to discuss the properties of the iterative minimum distance algorithm (IMDA) and quadriculation algorithm (QA) to obtain closed polygons of wildfire perimeters based on images of consumed vegetation in a landscape. This section summarizes the QUIC-Fire output data used for the evaluation of the IMDA and QA. The QUIC-Fire output data consist of images of the fuel densities over a landscape at ground level (below 10m) at different time stamps (100s, 300s, 500s, 700s, 900s, 1100s) as a prescribed burn or wildfire progresses. The images of the QUIC-Fire output data are given in Figure 1.



(a) 100s (b) 300s (c) 500s

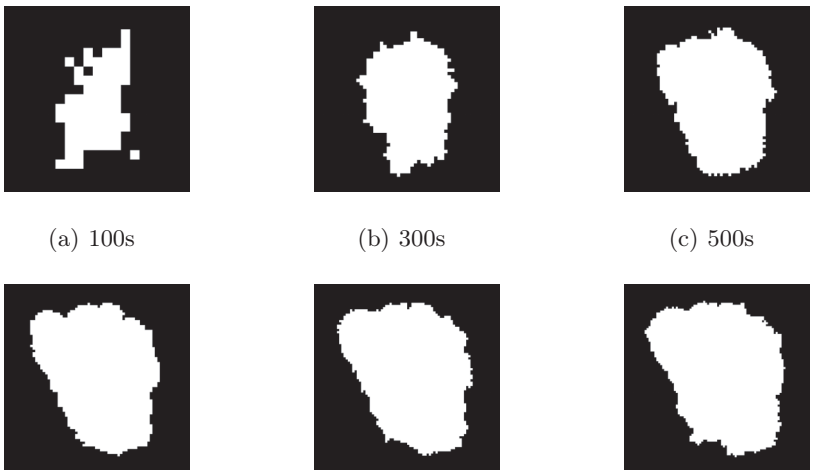
(d) 700s (e) 900s (f) 1100s

Fig. 1. Fuel densities at different time stamps after the wildfire begins.

From Figure 1, it can be observed that as time increases, the dark blue area with near zero fuel density becomes larger, which means the fuels are consumed

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_44

and the wildfire is spreading. The burn area of the wildfire at different time stamps can then be detected by comparing the difference of the corresponding fuel densities and the fuel density before the wildfire starts, leading to the black/white images given in Figure 2.



(a) 100s (b) 300s (c) 500s

(d) 700s (e) 900s (f) 1100s

Fig. 2. Burn area data at different time stamps after the wildfire begins. The white area is the burn area, and the black area is the unburned area. The scales of the six plots are selected differently for a better view.

3 Polygon algorithms for wildfire perimeters

3.1 Image data

The burn area at six different time stamps are illustrated in Figure 2. The white area represents the burn area with $y = 1$, and the black area represents the unburned area with $y = 0$, where

$$y_{i,j} = f([i,j],b) = \begin{cases} 0, & \text{if } b = 0 \\ 1, & \text{if } b > 0 \end{cases} \tag{1}$$

In (1), $[i,j]$ is a vector providing the position information of the target pixel in the image, and b is the absolute difference value between the fuel densities at the current time stamp and before the wildfire starts. The variable y is used to describe the area (burn area or unburned area) the pixel (at $[i,j]$) belongs to.

To better cover all possible situations of wildfire and illustrate the performances of IMDA and QA, one of the burn area outputs of Figure 2 has been

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_44

Fig. 3. Modified output of the QUIC-Fire with extra rectangular burn area. The white area represents the burn area, and the black area represents the unburned area.

increased in complexity by adding a separate (rectangular) burn area, and removing part of the original burn area, as indicated in Figure 3. The additional burn area is added to verify if both the IMDA and QA can recognize multiple wildfire perimeters within the data of Figure 3.

3.2 Quadriculation algorithm

The first method of finding an ordered set of vertices of a closed polygon around a burn area is the quadriculation algorithm (QA). Inspired by fire simulation tool FARSITE [5], the QA solves the problem in two main steps of division and union. Due to the fact that the minimum unit of a rasterized burn area image is a pixel, QA quadriculates the target image into four squares or rectangles recursively until all pixels in one square or rectangle have same value y. The process is illustrated on a simple example in Figure 4(a). It can be observed that after the first division, only the pixels in the right-top square of the image have the same value ($y = 0$). Therefore, another quadriculation is needed. The second division should be applied on left-top, left-bottom, right-bottom squares because pixels have different values $y = 1$ and $y = 0$ in these three squares, and no division should be applied on the right-top square.

After the recursive division, the adjacent squares or rectangles with same value y should be joined, and the perimeter of the polygon in Figure 4(a) can be obtained in Figure 4(b). With the precision of one meter for QUIC-Fire, the size of each cell is one meter times one meter. Therefore, the polygon obtained by QA can be accurate enough to describe the wildfire perimeter. The process of QA is summarized in Algorithm 1.

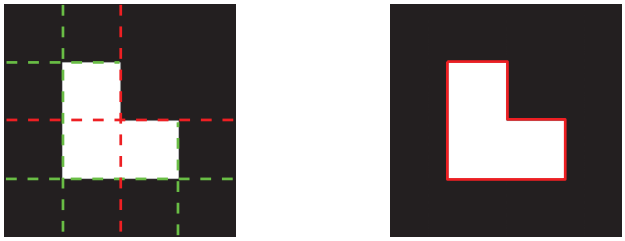
Algorithm 1 QA

Input: Fire image

Output: Polygons representing wildfire perimeters

1: Recursively quadriculate the image into four squares or rectangles until all pixels in one square or rectangle have same value y.
2: Join the adjacent squares or rectangles until the pixels in adjacent squares or rectangles have different value y.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_44



(a) Recursive division. (b) Union.

Fig. 4. Division and union in QA. The dashed red and green lines represents the first and second division respectively. The red solid line represents the polygon of the wildfire perimeter. The white and black area are with $y = 1$ and $y = 0$ respectively.

The QA is known to take quite some computation time due to two main steps of division and union that scales up as the image size increases. It would be beneficial to have an algorithm that can also handle large images with multiple burn areas. The proposed algorithm is presented in the next section.

3.3 Iterative minimum distance algorithm (IMDA)

Preparatory work The IMDA solves the problem of finding an ordered set of vertices of a closed polygon around a burn area by selecting and ordering the set of unordered boundary points. First, a standard image edge detection algorithm is applied to Figure 3 to acquire the boundary points. The boundary points are detected by comparing the value $y_{i,j}$ of the target pixel with its surroundings. An abrupt change in the y value of the pixel expressed by

$$\begin{aligned} |y_{i-1,j} - y_{i,j}| \neq 0 \text{ and } y_{i-1,j} = 1, \\ \text{or } |y_{i,j} - y_{i,j-1}| \neq 0 \text{ and } y_{i,j-1} = 1, \\ \text{or } |y_{i+1,j} - y_{i,j}| \neq 0 \text{ and } y_{i+1,j} = 1, \\ \text{or } |y_{i,j} - y_{i,j+1}| \neq 0 \text{ and } y_{i,j+1} = 1, \end{aligned}$$

and the pixel at i, j can be regarded as a boundary point. Due to the fact that the precision of the QUIC-Fire data can be as small as one meter, the edge detection achieves a resolution of one meter.

Naive minimum distance To motivate the IMDA, first consider the simplest method for the rearrangement of the unordered vertices or boundary points: choosing an arbitrary starting point and find the closest point to the previous selected point. In this native minimum distance (NMD) check, an important requirement is to avoid a self-intersection of the polygon.

With the set of the unordered boundary points B, the starting point b_1 is first selected arbitrarily. Then, remove b_1 from the set B, and find a new point

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_44

b_v ($v > 1$) with the minimum distance to b_1 in B. If the distance between the last two selected points b_{v-1} and b_v, where $v \geq 3$, is larger than the distance from b_{v-1} to b_1, b_{v-1} is connected to b_1 directly to produce a closed polygon. To ensure there is no problem of self-intersection, the NMD checks whether the line segment $b_{v-1}b_v$ intersects with any previous created line segments. If there exists an intersection, the point b_{v-1} is deleted and connect $b_{v-2}b_v$. This process iterates until no intersection exists.

During the process of finding b_v, two or more points can be found with same distance to the previous selected point (multi-choice situation). To solve this problem, each choice will be stored and the corresponding closed polygon is recorded. The polygon with the largest number of vertices is picked as an optimal choice because more vertices means more detailed information. If there are multiple polygons with same number of vertices, more constraints such as the area of the polygon, can be added to select the optimal polygon.

The main problem for the NMD check is that for each multi-choice situation, two or more complete polygons that are generated also need to be stored for comparison purpose. Storing and comparing polygons may be an computationally expensive process, especially when the numbers of boundary points and multi-choice situations increase. This problem is illustrated in a simple case of Figure 5. It can be observed that the red polygon better describes the burn area than the cyan dashed polygon, and the only difference between these two polygons is located inside the green dashed rectangle in the figure.

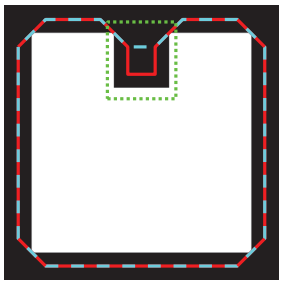


Fig. 5. Two possible polygons after removing self-intersections (red line and cyan dashed line). The green dashed rectangle shows the two-choice difference.

Next to storing and comparing multiple polygons, the NMD check cannot deal with the case when a wildfire has multiple disjoint burn areas to create multiple wildfire perimeters. These problems lead to a modification of the NMD check and result in the actual IMDA.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_44

Computation of ordered vertices of the closed polygon In the computation of ordered vertices of the closed polygon in IMDA, one initial main polygon is first obtained by arbitrarily choosing a point in the multi-choice situation. All left points are used to modify the initial main polygon or create a new isolated polygon. It is still assumed that all the unordered boundary points can be used only once, but with one more constraint: the largest distance between two adjacent boundary points should be smaller than $d = \sqrt{2}$ due to point-to-point pixel distances. Following this distance observation, there are two main steps in IMDA: the first step is to obtain an initial main polygon, and the second step is to modify the obtained polygon and decide whether there is an extra isolated polygon. The logic of each step is described as follows.

For the first step, an arbitrary starting point b_1 is selected from the set B of the unordered boundary points to be the first point of the set P that is used to restore the ordered vertices of the polygon of a wildfire perimeter. The point with the minimum distance to the previously selected point in P is chosen from B and added to P one by one. If there are multiple points with the same minimum distance to the previously selected point, the first point in order is selected. During the selection, if no other points in set B have the distance smaller than d with respect to the last point in P, the distance from the last point in P to the starting point b_1 is checked. If the distance is smaller than d, a closed polygon is created. On the contrary, if the distance is larger than d, it means the current trajectory is not correct. Therefore, the last point in P needs to be deleted and moved to a different set B_c so that this point can be reused again and ordered correctly. Repeat deleting the last point in P and move it to set B_c until a point with a distance smaller than d to the updated last point of P can be found in B, or the updated last point of P has the distance smaller than d to the starting point b_1. The first step is finished by creating an initial closed polygon.

With all the points moved from B_c to B, and clearing the set B_c, the second step is initiated by finding the nearest point in B to any vertex in P, if the distance is larger than d, it means no improvement can be achieved by the initial main polygon, and an isolated polygon exists. Hence, the first step should be repeated for the updated B to create a new initial polygon. If there exists a point in B with a distance to the nearest vertex in P smaller than d, it means the initial closed polygon can be updated. Based on closest vertex in P as the first point of the trajectory P_c, the nearest point from set B to the last point in P_c is found. If the distance from the newly detected point in B to the last point in P_c is smaller than d, then add the newly detected point to the set P_c. If no more points in B has the distance smaller than d to the last point of P_c, find the closest point in P to the last point in P_c. If the distance from the closest point in P to the last point in P_c is smaller than d, add the detected closest point in P to the set P_c. If the distance is larger than d, delete the last point in P_c, and add it to B_c until the distance from the closest point in P to the last point in P_c is smaller than d. Then add the detected closest point in P to the set P_c.

One important thing to note here is that the first point and the last point in P_c should be different from each other. Based on the first point and the last point

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_44

of P_c, add P_c to the initial created polygon. If the previously created polygon has other vertex between the first point and the last point of P_c, which means connecting P_c to P will lead to the deletion of previously selected vertices. Then, whether connecting P_c to P depends on whether connecting P_c will increase the area of the polygon. If connecting P_c to P can increase the area of the polygon, P_c is connected to P and replace the corresponding part selected in the first step. Otherwise, keep P as it is. Iterate this process until there is no point left in B and B_c. The logic process of the IMDA is summarized in Algorithm 2.

Algorithm 2 IMDA

Input: Unordered boundary points B and threshold value d.
Output: Polygons representing wildfire perimeters

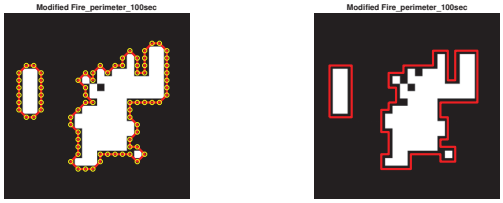
1: Pick the arbitrary starting point b_1 in B, and delete b_1 in B.
2: Find a closed polygon P based on finding the point with the minimum distance that is smaller than d to the previously selected point.
3: Find a trajectory P_c when the distance from any point in B to P is smaller than d.
4: If adding P_c to P will not result in the deletion of the previously selected point in P, add P_c to P.
5: If adding P_c to P will result in the deletion of the previously selected point in P, P_c is added to P when it increases the area of the polygon.
6: Iterate steps 3-5 until no points in B have distance smaller than d to P.
7: Repeat the above steps if there are multiple polygons.

4 Numerical Results

IMDA and QA are applied to the modified burn area data of Figure 3 to verify the detection of multiple fire perimeters. The resulting closed polygons created by IMDA and QA are shown in Figure 6. It is clear that both IMDA and QA produce the two distinct fire perimeters, but it can also be observed that IMDA provides slightly tighter polygons around the burn area as the polygons are not restricted to horizontal and vertical lines as in QA.

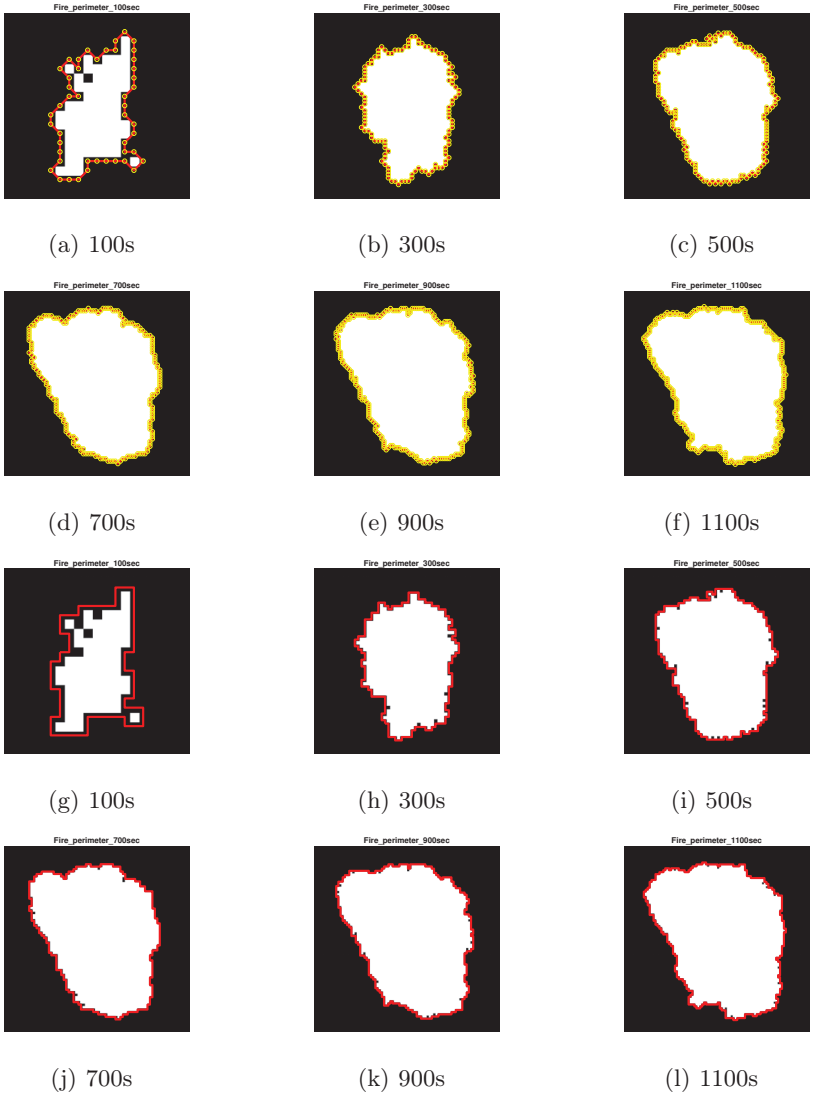
To further compare the performance of IMDA and QA, the algorithms are applied to the burn area data of at the six different time stamps of Figure 2. The visual results are summarized in Figure 7 with the same conclusion: both IMDA and QA produce correct results, but IMDA provides slightly tighter polygons. The more telling observations come from Table 1, where it can be seen that the computation time of IMDA scaled favorably compared to QA as the image size and the burn area of the wildfire perimeter increases. As reference for the computation time, all calculations were performed on an Intel Core i7-7500U CPU with 16 GB RAM.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_44



(a) IMDA (b) QA

Fig. 6. polygons of the wildfire perimeter (red lines). Burn area (white), unburned area (black), detected boundary points (yellow circles).



(a) 100s (b) 300s (c) 500s

(d) 700s (e) 900s (f) 1100s

(g) 100s (h) 300s (i) 500s

(j) 700s (k) 900s (l) 1100s

Fig. 7. Polygons of the wildfire perimeter (red lines). Burn area (white), unburned area (black) and detected boundary points (yellow circles).

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_44

	100s	300s	500s	700s	900s	1100s
IMDA	0.019s	0.019s	0.021s	0.020s	0.023s	0.024s
QA	0.041s	0.132s	0.185s	0.205s	0.362s	0.352s

Table 1. Computation time of IMDA and QA

5 Conclusions

This paper compares two algorithms, iterative minimum distance algorithm (IMDA) and quadriculation algorithm (QA), to obtain the closed polygons that parametrize wildfire perimeters. The IMDA is based on continually connecting two closest points in the set of unordered boundary points determined by conventional image edge detection. A threshold value is set up to assist in determining whether two points are closely located. From a completely different point of view, the QA creates a polygon by recursively dividing the raster image into indivisible rectangles, where all the internal pixels of the rectangles have the same color. Then, the QA merges adjacent rectangles that have the same color. Using simulation data produced by QUIC-Fire that consist of raster images of consumed vegetation in a landscape, the performance of IMDA and QA is compared. Although the logic of IMDA is more complicated, IMDA produces slightly tighter polygons around the burn area compared to QA, as the polygons are not restricted to horizontal and vertical lines in the image resolution. Moreover, the computation time of IMDA scaled favorably compared to QA as the image size and the burn area of the wildfire perimeter increase.

References

1. Achtemeier, G.L.: Field validation of a free-agent cellular automata model of fire spread with fire–atmosphere coupling. International Journal of Wildland Fire **22**(2), 148–156 (2012)
2. Cheney, N., Gould, J., Catchpole, W.: The influence of fuel, weather and fire shape variables on fire-spread in grasslands. International Journal of Wildland Fire **3**(1), 31–44 (1993)
3. Dickinson, M.B., Hudak, A.T., Zajkowski, T., Loudermilk, E.L., Schroeder, W., Ellison, L., Kremens, R.L., Holley, W., Martinez, O., Paxton, A., et al.: Measuring radiant emissions from entire prescribed fires with ground, airborne and satellite sensors–rxcadre 2012. International Journal of Wildland Fire **25**(1), 48–61 (2015)
4. Fang, H., Srivas, T., de Callafon, R.A., Haile, M.A.: Ensemble-based simultaneous input and state estimation for nonlinear dynamic systems with application to wildfire data assimilation. Control Engineering Practice **63**, 104–115 (2017)
5. Finney, M.A.: FARSITE, Fire Area Simulator – model development and evaluation, vol. 4. US Department of Agriculture, Forest Service, Rocky Mountain Research Station (1998)
6. García, N.L.F., Martínez, L.D.M., Poyato, Á.C., Cuevas, F.J.M., Carnicer, R.M.: Unsupervised generation of polygonal approximations based on the convex hull. Pattern Recognition Letters **135**, 138–145 (2020)
7. Gollner, M., Trouve, A., Altintas, I., Block, J., de Callafon, R., Clements, C., Cortes, A., Ellicott, E., Filippi, J.B., Finney, M., et al.: Towards data-driven operational wildfire spread modeling: A report of the nsf-funded wifire workshop. Tech. rep. (2015)

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_44

8. Graham, R.L., Yao, F.F.: Finding the convex hull of a simple polygon. Journal of Algorithms **4**(4), 324–331 (1983)
9. Linn, R., Reisner, J., Colman, J.J., Winterkamp, J.: Studying wildfire behavior using firetec. International journal of wildland fire **11**(4), 233–246 (2002)
10. Linn, R.R., Goodrick, S.L., Brambilla, S., Brown, M.J., Middleton, R.S., O'Brien, J.J., Hiers, J.K.: QUIC-fire: A fast-running simulation tool for prescribed fire planning. Environmental Modelling & Software **125**, 104616 (2020)
11. Mandel, J., Bennethum, L.S., Chen, M., Coen, J.L., Douglas, C.C., Franca, L.P., Johns, C.J., Kim, M., Knyazev, A.V., Kremens, R., et al.: Towards a dynamic data driven application system for wildfire simulation. In: International Conference on Computational Science. pp. 632–639. Springer (2005)
12. Mandel, J., Chen, M., Franca, L.P., Johns, C., Puhalskii, A., Coen, J.L., Douglas, C.C., Kremens, R., Vodacek, A., Zhao, W.: A note on dynamic data driven wildfire modeling. In: International Conference on Computational Science. pp. 725–731. Springer (2004)
13. Manzano-Agugliaro, F., Pérez-Aranda, J., De La Cruz, J.: Methodology to obtain isochrones from large wildfires. International journal of wildland fire **23**(3), 338–349 (2014)
14. Rothermel, R.C.: A mathematical model for predicting fire spread in wildland fuels, vol. 115. Intermountain Forest and Range Experiment Station, Forest Service, United ... (1972)
15. Srivas, T., Artés, T., De Callafon, R.A., Altintas, I.: Wildfire spread prediction and assimilation for farsite using ensemble kalman filtering. Procedia Computer Science **80**, 897–908 (2016)
16. Stow, D.A., Riggan, P.J., Storey, E.J., Coulter, L.L.: Measuring fire spread rates from repeat pass airborne thermal infrared imagery. Remote sensing letters **5**(9), 803–812 (2014)
17. Subramanian, A., Tan, L., de Callafon, R.A., Crawl, D., Altintas, I.: Recursive updates of wildfire perimeters using barrier points and ensemble kalman filtering. In: International Conference on Computational Science. pp. 225–236. Springer (2020)
18. Tan, L., de Callafon, R.A., Block, J., Crawl, D., Altıntaş, I.: Improving wildfire simulations by estimation of wildfire wind conditions from fire perimeter measurements. In: International Conference on Computational Science. pp. 231–244. Springer (2021)
19. Tan, L., de Callafon, R.A., Block, J., Crawl, D., Çağlar, T., Altıntaş, I.: Estimation of wildfire wind conditions via perimeter and surface area optimization. Journal of Computational Science p. 101633 (2022)
20. Valero, M., Rios, O., Pastor, E., Planas, E.: Automated location of active fire perimeters in aerial infrared imaging using unsupervised edge detectors. International journal of wildland fire **27**(4), 241–256 (2018)
21. Zajkowski, T.J., Dickinson, M.B., Hiers, J.K., Holley, W., Williams, B.W., Paxton, A., Martinez, O., Walker, G.W.: Evaluation and use of remotely piloted aircraft systems for operations and research–rxcadre 2012. International Journal of Wildland Fire **25**(1), 114–128 (2015)

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_44