Visualization of Direction of Wildfire Progression from Remote Sensing Images

Yu Zhao

March 28, 2022

1 Introduction

Wildfires have increased in their frequency and severity due to the hotter and drier conditions in many areas caused by human-induced climate change.[1][2] To help with the management of these wildfires, there is a significant need for better automatic detection of early and active fires and automatic burned area mapping. Owing to the emerging of remote sensing technology, automatic monitoring and detecting the wildfire becomes available by analysing the satellite images using deep learning. Consequently, active fire points could be plotted on a map to indicate the region affected by the active wildfire.

However, even though the progression of the wildfire can be indirectly interpreted by visualizing active fire points from different dates, it is hard for decision makers to analyse the trend of the wildfire progression from merely active fire points. There are multiple existing works tried to solve this problem. In [3], a spectral color hue-based representation of wildfire progression is most appreciated by the decision makers with limited time frame. In [4], the author introduced a visualization of wildfire progression similar to previous work. As the same time, it provides a modelling of potential affected region by calculating the distance to residential areas. However, I argue this spectral color visualization does not clearly show the direction of the progression which helps decision makers like fire fighters to make crucial decisions on prioritised regions.

In summary, the main objective of this research is to design a way of visualizing the direction of wildfire progression which generated from remote sensing images. The results show that by using clustering algorithm, direction of wildfire progression can be visualized by connecting the center of burned area in previous timestamp to each center of clusters. Moreover, by visualizing current direction of the wildfire, the proposed method also helps to predict the future progression directions which provides valuable information for decision makers.

2 Dataset Inspection

The dataset consists of wildfire images from 10 different days. There are in total 3 different sizes of wildfires which corresponding to small, medium and large wildfires. The remote sensing images are genereated from Visible Infrared Imaging Radiometer Suite(VIIRS) which is embedded in Suomi-NPP satellite. The spatial resolution is 375 m which means one pixel covers 375m by 375m square of land. Active fire points are generated from deep learning models which automatically classify the pixels within the remote sensing images as active fire pixels.

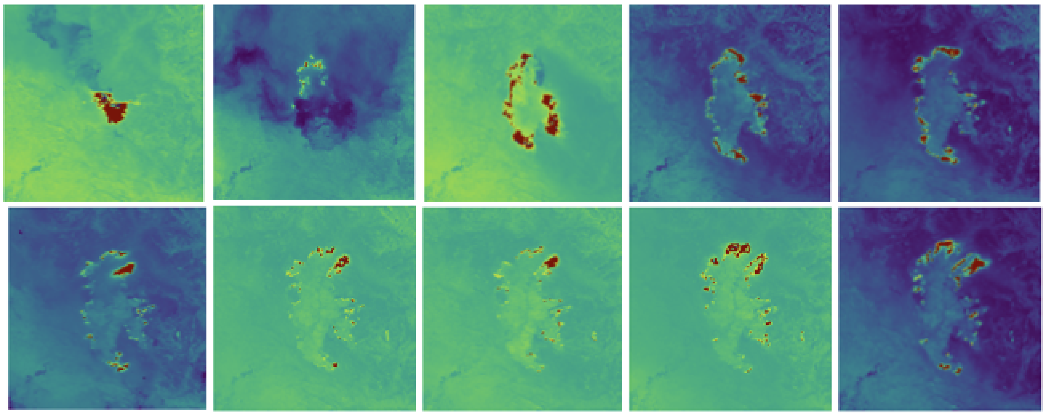

Figure 1: Creek Fire from Wildfire progression dataset, the active fire points are overlaid on original remote sensing images. Red dots in the images are active fire pixels.

As shown in Fig 1, it can be clearly observed that the wildfire in selected study area starts at the center of the image and gradually ignite connecting biomass. However, the intensity of the wildfire is different spatially. Priorities should be given to regions which have larger fire that could cause severe damage to the land covers. Thus, it is crucial to know in which direction the wildfire progresses.

3 Methodology

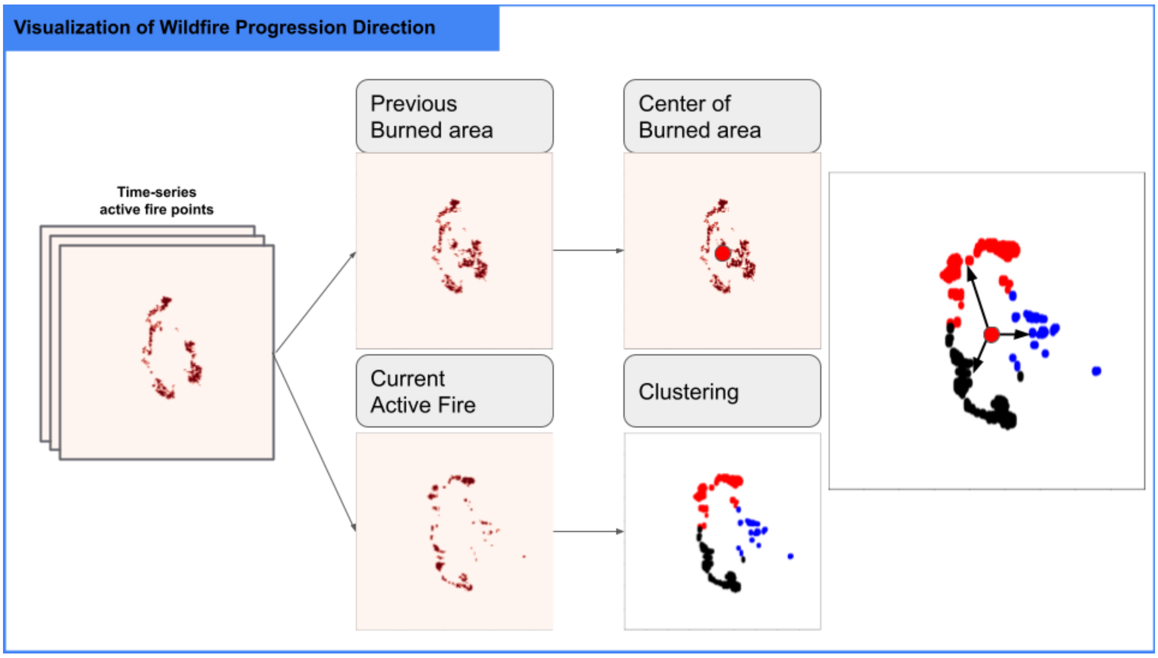

Figure 2: Workflow of visualization on Wildfire progression direction

As shown in Fig 2, there are two components within the workflow. At first, if we assume current timestamp is $t \in [0, 10)$, the center of current burned area is calculated using stack of all previous images from $[0, t)$. This point is used as the start point of the direction arrows. To find the end points of the arrows, clustering algorithm is applied to all the active fire points at current timestamp t. In this proposed workflow, K-means algorithm[5] is used to find center points of the clusters of active fire points. K-Means algorithm randomly initialize the start point from the dataset based on the preset number of clusters provided and classify the points based on the similarities of coordinates to the start point. Consequently, active fire points are labeled based on their spatial location. Moreover, the center point of the cluster is also obtained from the algorithm by constantly adjust its position based on labeled dataset. By connecting the start point and the center points of the cluster, the arrow is created to mark the wildfire progression direction, as it represents the direction from the center of previous burned area to the center of the most severe active fire at current timestamp.

4 Results and Analysis

4.1 Quantitative results

The quantitative results focus on the validation on the effectiveness of prediction future progression, it is evaluated by comparing the direction assigned at time T and the direction calculated at time T+1 to show how well the direction at current timestamp forecasts the wildfire progression direction in the future. More specifically, assume the progression represented by vectors at time t is V_t with length k and the progression direction at t+1 is V_{t+1}, the metrics based on cosine similarity of two sets of vectors can be fomulated as below:

$$\frac{1}{k}\sum_{i=1}^{k} Cos\_Similarity(m_i, n_i) = \frac{1}{k}\sum_{i=1}^{k} \frac{m_i \cdot n_i}{||m_i|| \cdot ||n_i||}, m_i \in V_t, n_i \in V_{t+1}$$

From the formula above, it can be observed that when all the directions at time t is strictly aligned with all the direction at time t+1, the above metric has value equals to 1. When all the current directions are opposite to the direction in the future, the value equals to -1. And when both directions perpendicular to each other, the value becomes 0. Thus, if the metric has high score close to 1, it means that current assigned direction is very likely to be the progression direction in the future. From the quantitative results

Table 1: Cosine Similarities of testing sites

	Creek Fire	Dixie Fire	Lytton Fire
Cosine Similarity	0.773	0.875	0.613

of all the testing sites, it can be observed that large scale fire and medium scale fire achieves above 0.7 as the cosine similarity, while small scale fire only has 0.6. The scale and severity of the fire affects the moving direction of the wildfire since strong wildfire usually progress along the direction of the wind and the direction where has more biomass. Thus, the moving direction of small scale wildfire tends to be less deterministic. In summary, using current moving direction to forecast moving direction in the future gives optimistic results for medium and large scale wildfires.

4.2 Qualitative results

The qualitative results focus on how the proposed method helps to improve the illustration of the wildfire progression direction by comparing the proposed vector based method with the hue-based method presented in [3].

4.2.1 Study area 1: Creek Fire, California, US

Creek Fire, which caused 379,895 hectares of forest burned, is the second largest wildfire that has happened in the fire season of 2020. The fire starts at 2020-09-05 and first 10 days progression is tested with the network and the results are overlaied on Band I4 images. The spread speed of this fire is much higher than the other two wildfires, which caused more active fire pixels in the early stages.

As shown in Figure 3, the last column shows the wildfire progression direction with cluster number equals to 3. It can be observed that the wildfire spreads to all the directions, that 3 vectors can pretty well represent the moving direction of the wildfire. For example, at row 6 and 7, the wildfire stops in the southwest direction and the vectors correctly direct to northeast direction. Although, I argue for wildfire which spreads to all the directions, the number of components should be set larger to better represent the moving direction. As the trade-off, the calculation time will also be higher.

Qualitatively, by comparing column 3 and column 5, hue-based method use light red to mark old active fire and dark red to mark latest active fire. Although the spreading direction can be visually inspected, it is not as direct as vector-based method.

4.2.2 Study area 1: Dixie Fire, California, US

Dixie Fire, which caused 389,837 hectares of forest burned, is the largest wildfire that has happened in the fire season of 2021. Compared to Creek Fire, the emerging of Dixie fire is slower. So for the first seven days, the Dixie fire can be considered as a medium size wildfire.

Different from creek fire, dixie fire as shown in Figure 4 majorly moving to northeast direction, this makes the forecast much simpler than creek fire. From the qualitative results, it can be observed that the moving direction at t can very well represent the moving direction at t+1.

4.2.3 Study area 1: Lytton Fire, British Columbia, Canada

Lytton Creek fire , one of the most devastating fire in British Columbia , burned 83,671 hectares including the town of Lytton. Majority of the burned area is resulting in significant economic impact. Compared to two other wildfire, Lytton fire is smaller in size, which makes it increasingly difficult to detect, and prone to have false positive and false negative predictions.

As discussed in quantitative results, the forecasting results of Lytton Fire is less significant than two other wildfires because of its scale. Although, when we only looking at qualitative results, the spreading direction can be very lucidly presented by vectors compared to hue-based method. Hue-based visualization method does not provide very big contrast between active fire pixels at different days because of the slow progression of the wildfire and it almost stop in fifth day. But through vector-based method, the moving direction can still be tracked.

5 Conclusion

In summary, in this work, a vector-based visualization method for wildfire progression direction is proposed. Based on the density of the active fire points, K-means clustering algorithm is applied to the given data at first. From visual inspection, the clustering result very well represents the spatial allocation of the active fire pixels. Then, by connecting the centroid of the burned area and the centers of active fire points clusters, the arrow which represents the progression direction of the wildfire can be obtained.

Qualitatively, the proposed method can very well represents the active fire spreading directions compared to hue-based method in all testing sites. And though quantitative evaluation on the forecasting ability, the direction vectors show promising results when forecasting the spreading direction for large and medium wildfire using current spreading direction. For future improvement, one possible direction is to automatically adjust the number of clusters according to the number of pixels for one wildfire to make it more scalable.

References

[1] Yue, X. et al, "Ensemble projections of wildfire activity and carbonaceous aerosol concentrations over the western united states in the mid-21st century," *Atmospheric environment (Oxford, England : 1994)*, vol. 77, pp. 767–780, 10 2013.

[2] Teague, B., Pascoe, S. and Mcleod, R., "The 2009 victorian bushfires royal commission final report," 2010.

[3] Cheong, L. et al, "Evaluating the impact of visualization of wildfire hazard upon decision-making under uncertainty," *International Journal of Geographical Information Science*, vol. 30, pp. 1377 – 1404, 2016.

[4] Kim, T.H., Cova, T.J. and Brunelle, A.R., "Exploratory map animation for post-event analysis of wildfire protective action recommendations," *Natural Hazards Review*, vol. 7, pp. 1–11, 2006.

[5] MacQueen, J., "Some methods for classification and analysis of multivariate observations," 1967.

Appendix

```
import itertools
from copy import deepcopy
from math import sqrt

import matplotlib.pyplot as plt
import numpy as np
import scipy.signal
from sklearn.cluster import KMeans
from sklearn.metrics.pairwise import cosine_similarity


def get_direction_vector(x, y, fire_t):
    '''
    Args:
        x: x-coordinate of centroid of fire pixels
        y: y-coordinate of centroid of fire pixels
        fire_t: fire pixels at timestamp t

    Returns: List of direction vectors

    '''
    vectors = []
    for i in range(fire_t.shape[0]):
        for j in range(fire_t.shape[1]):
            if fire_t[i, j] > 0:
                length = sqrt(pow(i - x, 2)+pow(j - y, 2))
                vectors.append([(j - y), -(i - x),  length])
    return vectors

def evaluate(dir):
    '''
    Args:
        dir: List of direction vectors

    Returns: maximum cosine similarity between two direction vectors.

    '''
    def evaluate_between_each_day(dir1, dir2):
        combination = []
        length = len(dir1)
        digits = [i for i in range(length)]
        for j in itertools.permutations(digits, 3):
            combination.append(j)

        score_comb = 0
        for comb in combination:
            score = 0
            for k in range(length):
                score += cosine_similarity(np.array(dir1[k]).reshape(1,-1), np.
array(dir2[comb[k]]).reshape(1,-1))
            score_comb = max(score/length,score_comb)
```

```
        return score_comb
    score_avg = 0
    for i in range(1, len(dir)):
        score_avg += evaluate_between_each_day(dir[i-1], dir[i])
    return score_avg/(len(dir)-1)


if __name__=='__main__':
    # Name of testing sites
    fires = ['creek_fire', 'dixie_fire', 'lytton_fire']
    for fire in fires:
        array = np.load(fire+'.npy')
        stack = array[0, :, :]
        stack_hue = deepcopy(array[0, :, :])
        stack_hue[stack_hue==0]=np.nan
        dir = []
        for i in range(1, 10):
            fire_t = array[i, :, :]
            # Calculate centroid of the fire
            index = np.where(stack>0)
            if index[0].size != 0:
                x = int(index[0].mean())
                y = int(index[1].mean())

                vectors = get_direction_vector(x, y, fire_t)
            else:
                vectors = [[0, 0]]
            # Visualization
            plt.figure(figsize=(20, 4))
            plt.subplot(151)
            plt.axis('off')
            plt.imshow(fire_t,cmap='Reds')
            # plt.savefig('../plt/'+fire+'fire_t'+str(i)+'.png', bbox_inches='tight')
            plt.subplot(152)
            plt.axis('off')
            plt.imshow(stack,cmap='Reds')
            plt.subplot(153)
            plt.axis('off')
            plt.imshow(stack_hue,cmap='Reds')

            plt.subplot(154)
            plt.axis('off')
            num_components = 3
            spatial_data = []
            for l in range(fire_t.shape[0]):
                for p in range(fire_t.shape[1]):
                    if fire_t[l,p] != 0:
                        spatial_data.append([l,p])

            # Kmeans Clustering of fire pixels
            if len(spatial_data)==0:
                continue
```

```python
        else:
            spatial_data = np.array(spatial_data)
            cluster = KMeans(n_clusters=num_components, random_state=0).fit(spatial_data)
            cluster_center = cluster.cluster_centers_.astype(int)

        clustering_result = np.zeros((224,224))
        for k in range(len(spatial_data)):
            clustering_result[spatial_data[k][0], spatial_data[k][1]] = cluster.labels_[k]+1
        plt.scatter(spatial_data[cluster.labels_ == 0][:,1],-spatial_data[cluster.labels_ == 0][:,0], color='red')
        plt.scatter(spatial_data[cluster.labels_ == 1][:,1],-spatial_data[cluster.labels_ == 1][:,0], color='black')
        plt.scatter(spatial_data[cluster.labels_ == 2][:,1],-spatial_data[cluster.labels_ == 2][:,0], color='blue')
        plt.xlim([0, 224])
        plt.ylim([-224,0])
        plt.subplot(155)
        plt.axis('off')

        # Visualize direction vectors of active fire
        dir_day = []
        for j in range(num_components):
            arrow_x = cluster_center[j][1]-y
            arrow_y = -(cluster_center[j][0]-x)
            dir_day.append([arrow_x, arrow_y])
            plt.quiver(x, y, arrow_x, arrow_y, angles='xy', scale_units='xy', scale=1)
            plt.xlim([0, 224])
            plt.ylim([0, 224])
        dir.append(dir_day)
        plt.savefig('../plt/'+fire+'dir'+str(i)+'.png', bbox_inches='tight')
        plt.show()

        stack = np.logical_or(stack, fire_t)
        stack_hue[fire_t!=0]=i+1
        stack_hue[stack_hue==0]=np.nan
    print('Cosine Similarity of fire ' + fire+ ' {}'.format(evaluate(dir)))
```

Listing 1: Python Implementation

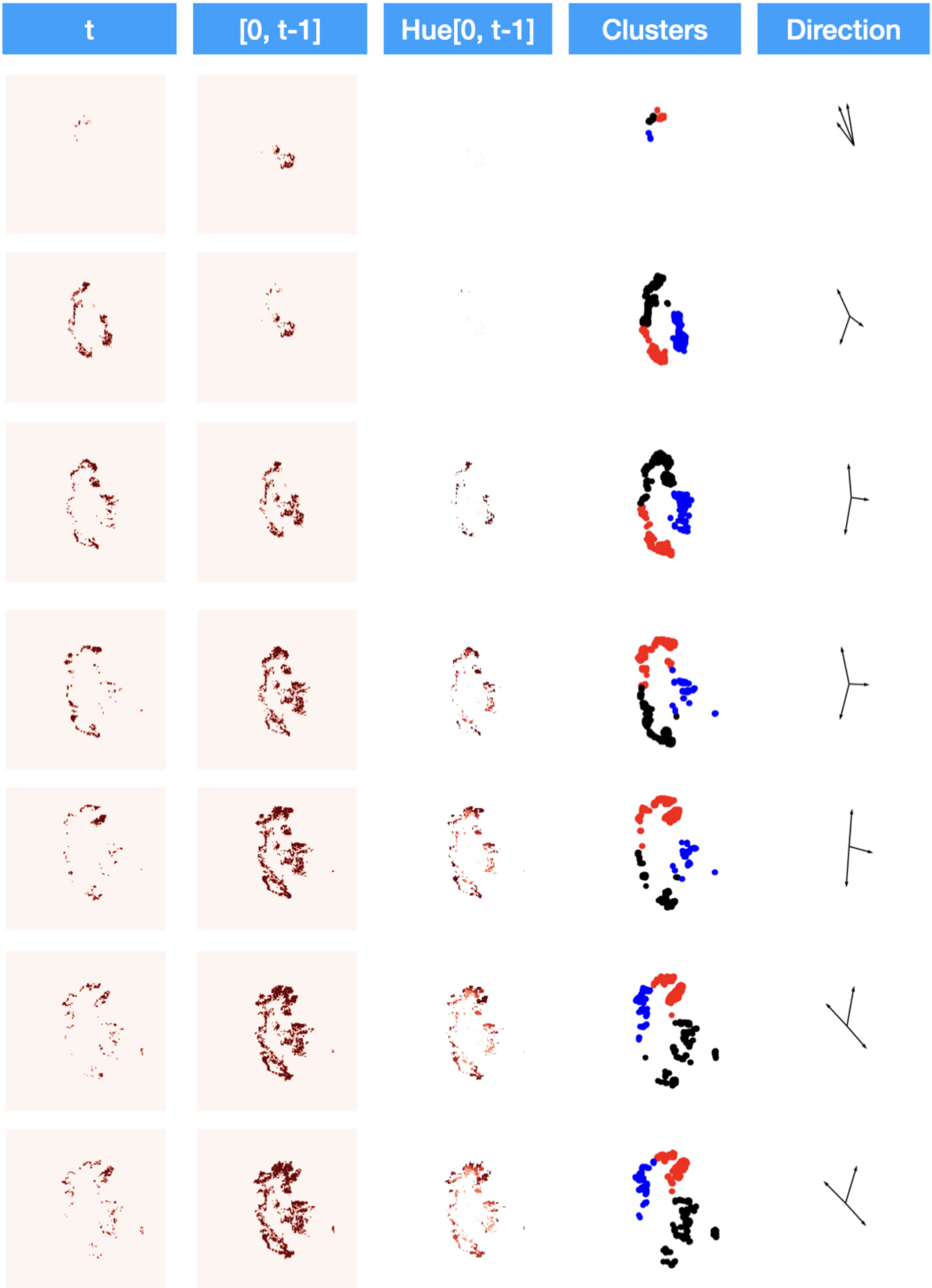

Figure 3: Qualitative results of creek fire from t=1 to t=7

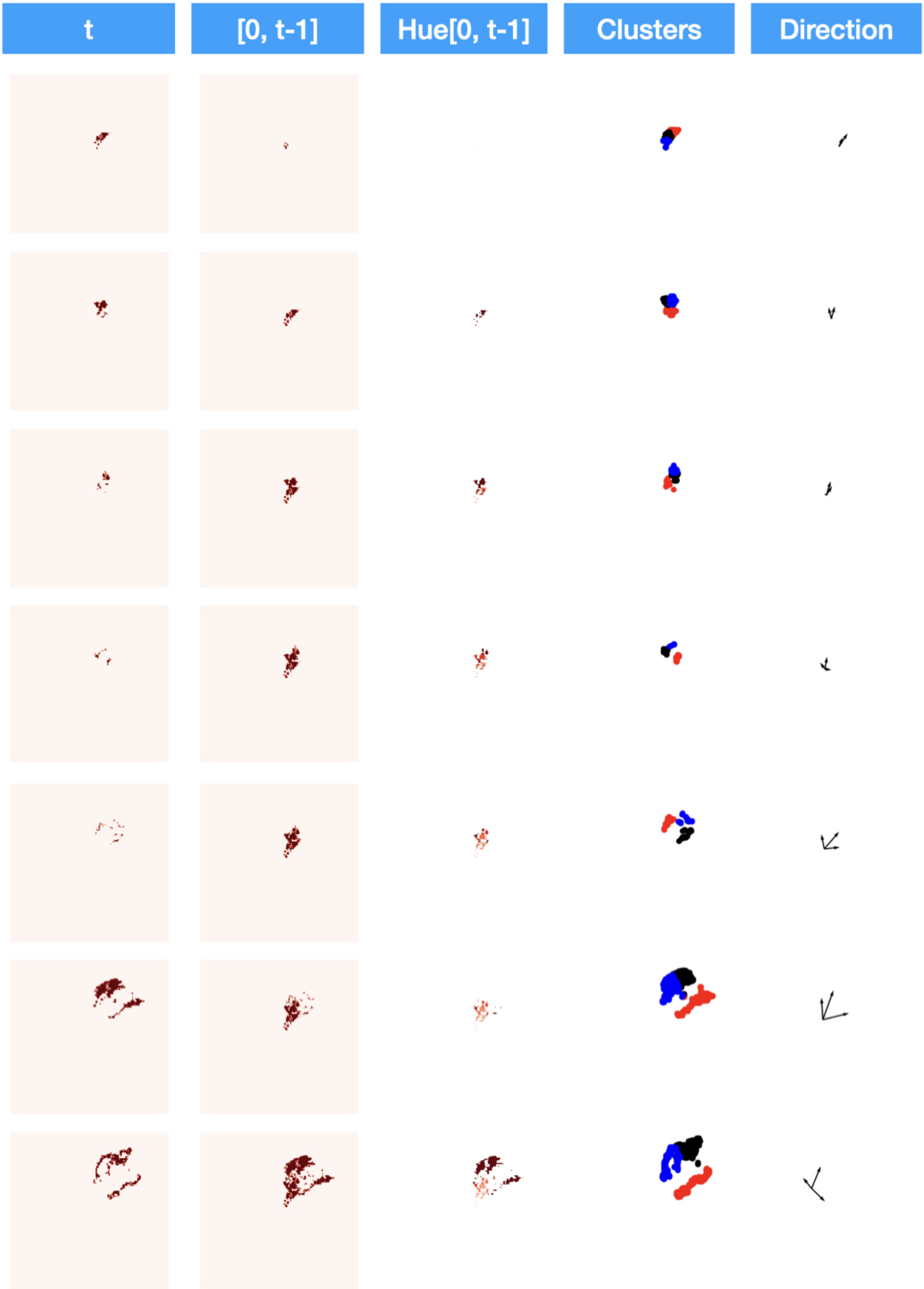

Figure 4: Qualitative results of dixie fire from t=1 to t=7

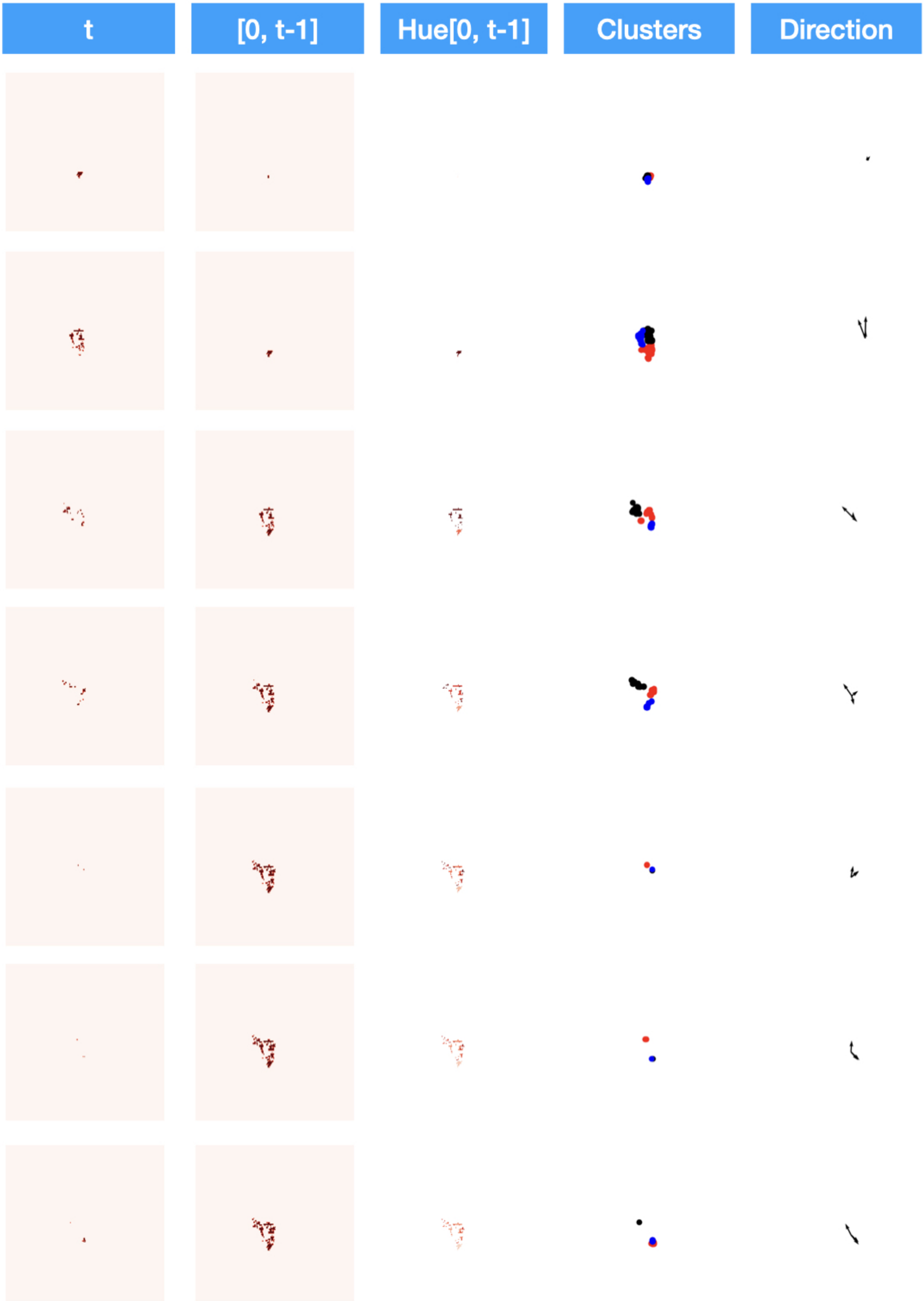

Figure 5: Qualitative results of lytton fire from t=1 to t=7