Improvement of Firebrand Tracking and Detection Software

Prohanov S.[1[0000-0002-4478-2249]], Kasymov D.[1,*[0000-0003-3449-788X]], Zakharov O.[1], Agafontsev M.[1[0000-0003-1383-7291]], Perminov V.[1[0000-0001-5393-1851]], Martynov P.[1[0000-0001-9489-1390]], Reyno V.[2], Filkov A.[3[0000-0001-5927-9083]]

[1] Tomsk State University, Tomsk, Russia
[2] V.E. Zuev Institute of Atmospheric Optics of the Siberian Branch of the Russian Academy of Science, Tomsk, Russia
[3] The University of Melbourne, Creswick, Australia
denkasymov@gmail.com

Abstract. Burning and glowing firebrands generated by wildland and urban fires may lead to the initiation of spot fires and the ignition of structures. One of the ways to obtain this information is to process thermal video files. Earlier, a number of algorithms were developed for the analysis of the characteristics of firebrands under field conditions. However, they had certain disadvantages. In this regard, this work is devoted to the development of new algorithms and their testing.

For this purpose, semi-field experiments were conducted using an apparatus for generating firebrands to obtain the necessary thermal video files. The thermograms were processed to create an annotated IR video base that was further used to test the detector and the tracker.

To detect firebrands in the thermograms, the Laplacian of Gaussian and Difference of Gaussians (DoG) algorithms were tested. To estimate the accuracy of detectors, an original approach involving the application of the F1 score was used. The analysis showed that both algorithms can provide the necessary accuracy for the detection of firebrands and are comparable in time and accuracy, but the DoG algorithm is easier controlled and implemented.

Different firebrand tracking algorithms have been developed and tested. In particular, a Hungarian algorithm-based tracker that can track firebrands between frames with high accuracy is implemented. The comparison of the algorithms showed that Hungarian algorithm-based trackers more accurately tracked the movement of particles.

Keywords: Algorithm, Detection, Firebrands, Tracking.

1 Introduction

In recent years, the number of wildland urban interface fires (WUI fires) has increased. The ignition of buildings in WUI areas is a serious international problem due to large fires in Australia, Greece, Portugal, Spain and the USA [1-4]. The main factors affecting the ignition of building materials and the spread of such fires are radiation and convective heat transfer, as well as firebrands that can accumulate on the roof and in the corners of buildings, on fences or find another way to penetrate into buildings and cause a fire [5-11].

In addition, burning and glowing firebrands produced in the fire front can be transported by wind and cause spot fires. WUI fires are expected to be a serious problem not only for the USA, Europe and Australia, but also for Russia.

At present there is a need in a quantitative understanding of the short distance spotting dynamics, namely the firebrand distribution within a distance from the fire front and how fires coalesce. The absence of such data is an obstacle to the development of fire hazard forecasting methods, as well as to the improvement of measures and recommendations for more efficient and effective work to prevent fires, control and extinguish ground forest fires in proximity to residential buildings.

To address this, a first version of custom software was developed in order to detect the location and the number of flying firebrands in a thermal image and then determine the temperature and sizes of each firebrand [12]. The software consists of two modules, the detector and the tracker. The detector determines the location of firebrands in the frame, and the tracker compares a firebrand in different frames and determines the identification number of each firebrand. However, used algorithms had certain disadvantages. In this regard, this work is devoted to the development of new algorithms and their testing using experimental data.

The non-contact IR diagnostic method based on modern high-speed IR cameras with high spatial resolution can estimate the temperature and size of particles, as well as the speed and trajectory of their movement. These data and the setup simulating the transfer of burning and glowing firebrands depending on the speed of air and their number [13-15] allow the software to be adjusted and verified.

2 Methods

2.1 Creation of an annotated video database

In 2015, a unique setup was designed and built to produce burning and glowing firebrands of various types, sizes, speeds and shapes [13, 16] (see Fig. 1).

Fig. 1. Setup for producing burning and glowing firebrands.

This setup designed by the authors has a number of distinctive features. A screw conveyer with an air smoke screen was mounted in the setup for a long continuous supply of fuels. The setup consists of three units, and its characteristics can be easily changed for different tasks [16]. The setup generating burning and glowing firebrands includes a unit for measuring temperatures and heat fluxes, a video camera and high-speed IR cameras (JADE J530SB and FLIR X6530sc) used as recording equipment for tracking of firebrands (see Fig. 2).

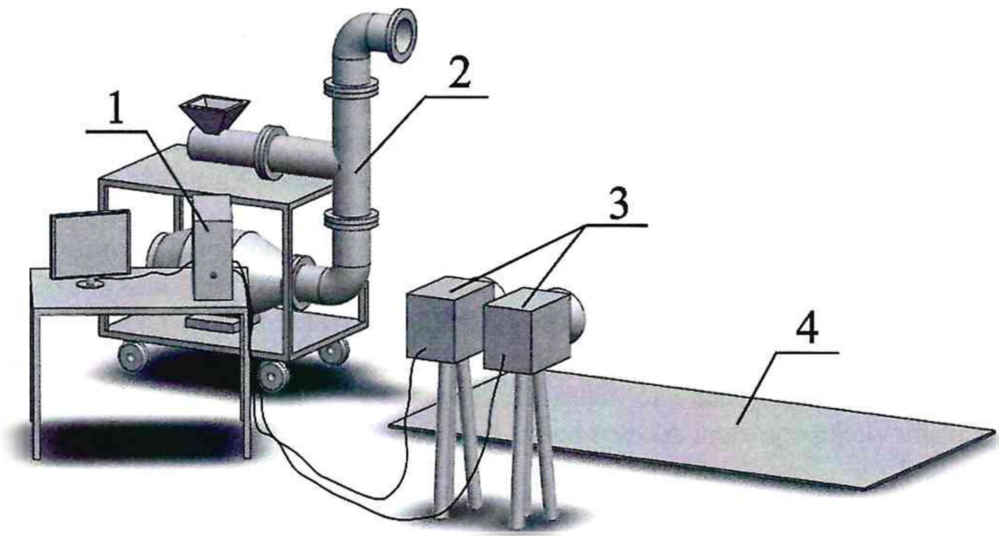

Fig. 2. Experimental setup: 1 is the data recording system, 2 is the producer of burning and glowing firebrands, 3 is the IR cameras, 4 is the test site.

The important characteristics of firebrands produced by wildland fires are the temperature, the distance of transport, and the flight of trajectory. IR cameras in combination with the firebrand producing setup successfully determine the desired characteristics.

A thermal imaging camera (JADE J530SB) containing a narrow-band optical filter with a 2.5 - 2.7 μm spectral interval and measuring the temperature in the range of 310 - 1500 C was used to determine the temperature of firebrands generated by the setup.

The thermal imaging camera had a matrix with a resolution of 320x240 pixels. A lens with a focal length of 50 mm was used for recording; the recording rate was 50 Hz. The optical filter and the lens had the factory calibration. The distance from the output of the particle generator to the thermal imaging camera was 8.7 m. The FLIR X6530sc thermal imaging camera with a spectral interval of 1.5 - 5.1 μm was used to determine the geometrical parameters of flying firebrands. The thermal imaging camera had a matrix with a resolution of 640x512 pixels. A lens with a focal length of 25 mm was used for recording; the recording rate was 50 Hz. The distance from the thermal imaging camera to the central plane of the output of the setup was 2.2 m, and the distance between the thermal imaging cameras was 0.4 m. The measured area was 1.7x1.35 m for the FLIR X6530sc and 0.42x0.32 m for the JADE J530SB.

Natural particles (pine bark and twigs) and wood pellets were used as firebrands. The size of the particles was selected in accordance with the data of field experiments which showed that particles of similar sizes prevailed during a surface fire in a pine forest [17–19]. Natural particles were made of pine bark (Pinus sibirica) and were 10×10, 15×15, 20×20, 25×25, 30×30 mm^2 in size and 5 mm in thickness. Pine twigs with a diameter of 2–4, 4–6, 6–8 and a length of 10, 20, 40 and 60 mm were also used. The diameter of wood pellets was 8 mm, the length of granules was varied from 30 to 50 mm. Each experiment had at least 3 repetitions.

2.2 Development of GUI

A graphical user interface (GUI) was developed to work with a software for detecting, tracking, and determining the characteristics of burning and glowing firebrands in a video using thermal imaging cameras.

To develop the GUI, available libraries of graphic elements that support working with the used software language python3 were considered and compared. In particular, the libraries Tkinter [20], pyGTK / PyGObject [21], wxPython [22], and pyQt5 [23] were considered.

Tkinter is a cross-platform library for developing a graphical interface, stands for the Tk interface and is an interface to Tcl/tk [24], is included in the Python language and does not require additional software installation. However, this library has a relatively small set of built-in widgets and rather poor integration with the desktop environment. The PyGTK/PyGObject library is used to develop cross-platform applications for GNU/Linux and Windows operating systems and write a quite compact code, but it requires additional software installation and the interface of the application under Windows OS is quite non-native. The wxPython library is the wrapper of the wxWidgets library [25]. It also can be used to develop cross-platform applications, but this library is quite complicated and has incomplete documentation. The choice was made in favor of the PyQt5 library, since it can work with the Qt cross-platform framework [26] which allows applications to be run under various operating systems (Windows, MacOS X, Linux).

To render the video frames, a free graphic component library Qwt [27] that works in conjunction with a library Qt was used. This library provides animation and scaling of data.

The graphical interface is used to open and visualize videos in the ASCII frame format.

Example of the used data format (ASCII):

```
General information :
File  G:\DRAGON_11_05_2018\Capture004_comp.ptw
Date  Sunday, March 05, 2018
Total frames 2789
Format  320 240
Radiometric data :
Calibration file G:\DRAGON_11_05_2018\Capture004_comp.exp
Unit  °C
Emissivity 0.95
BackGround temperature 22.00 °C
Transmission 100.00  %
Distance  2200.00 mm
Atmosphere temperature 26.00 °C
Housing temperature  22.70 °C
Pixel size  25.00 µm
Pixel pitch 30.00 µm
Focal length 50.00 mm
Aperture  2.00  F/#
Cut on  3.70  µm
Cut off 4.80  µm

Image Data :
Frame 0
Time  02:53:41,417
280.01  280.93  280.62  280.97  280.49  280.71  280.32
  281.01  280.49 …
…
282.10  282.32  282.40  282.36  282.32  282.71  282.32
  282.58  282.58 …

Image Data :
Frame 1
Time  02:53:41,437
…
```

2.3 Detector testing

To detect burning and glowing firebrands on the frame, various Gaussian convolution algorithms were tested. The algorithms are aimed at identifying pronounced areas. The Laplacian of Gaussian (LoG) and Difference of Gaussian (DoG) algorithms were tested.

The Gaussian of Laplacian algorithm (LoG) is based on the convolution (filtering) of an image using the Laplace operator:

$$G_\sigma(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(\frac{x^2+y^2}{2\sigma^2}\right)}, \tag{1}$$

where G is the Gaussian kernel, σ is the Gaussian parameter, x, y are spatial coordinates.

The Laplacian of Gaussian can be given by the formula

$$\Delta\left[\left(G_\sigma(x,y) * f(x,y)\right)\right] = LoG * f(x,y), \tag{2}$$

where $LoG = \frac{d^2}{dx^2} G_\sigma(x,y) + \frac{d^2}{dy^2} G_\sigma(x,y)$ is the Laplacian operator.

The Difference of Gaussian (DoG) algorithm is based on the two convolutions of the image with a Gaussian with a different parameter of σ.

$$g_{\sigma_1}(x,y) = G_{\sigma_1}(x,y) * f(x,y), \tag{3}$$

$$g_{\sigma_2}(x,y) = G_{\sigma_2}(x,y) * f(x,y), \tag{4}$$

where $G_{\sigma_1}(x,y)$ and $G_{\sigma_2}(x,y)$ are the Gaussian kernels.

The second step of the algorithm is pixel-by-pixel subtraction of the Gaussian images from each other.

$$g_{\sigma_1}(x,y) - g_{\sigma_2}(x,y) = DoG * f(x,y), \tag{5}$$

where $DoG = G_{\sigma_1} - G_{\sigma_2}$ is the Difference of Gaussian operator.

To evaluate the accuracy of the detectors, the F1 score was used [28]:

$$\mathrm{F1} = \frac{2\mathrm{TP}}{2\mathrm{TP}+\mathrm{FP}+\mathrm{FN}}, \tag{6}$$

where TP is the number of true positives, FP is the number of false positives and FN is the number of false negatives.

This metric takes into account two types of false positives: (i) when a background element is falsely detected (false operation), (ii) when a visible particle is not detected.

2.4 Tracker Testing

After detecting all particles in the frame, it is necessary to determine whether it is a new particle or it is in the previous frame and assign it a unique identification number as well. For this purpose, special particle trackers were developed.

In the previous version of the software, the tracker was based on the nearest neighbor search method (NSS) [29]. Each detection in the current frame, according to the selected metric, was compared to the detection in the next frame, all other detections were ignored. The advantages of this tracker are simple implementation, high operation speed, low memory consumption. However, the method has significant disadvantages, such as a high error, a strong dependence on the choice of the metric and, as a consequence, a low accuracy of operation.

Therefore, additional trackers were tested. A tracker based on the Hungarian algorithm [30] (Kuhn-Munkres algorithm) can track the movement of detection between frames with a high accuracy. The algorithm is applied to all pairs of frames; all detections in the first frame are compared to all detections in the next frame (except the absence of corresponding detection). The main stage of the algorithm is the construction of a cost matrix that, according to the algorithm, can find the optimal match between the considered detections. Unlike the nearest neighbor search algorithm, this algorithm compares all detections of particles between adjacent frames with each other and determines the optimal match, rather than compares the detections individually. This allows the algorithm to work with a higher accuracy. The distance between particles, the size and temperature of the particle were used as a metric for comparing particles in frames.

The Multiple Object Tracking Precision (MOTP) and Multiple Object Target Accuracy (MOTA) metrics [31] were used to evaluate the quality of the trackers.

The MOTP metric is calculated by the formula (7) and used to evaluate the positioning accuracy of the tracked particles.

$$MOTP = \frac{\sum_{i,t} d_i^t}{\sum_t c_t}, \tag{7}$$

where d_i^t is the distance between the ground truth, i.e. annotated particles and the particles predicted by the code with number i in the frame; c_t is the number of matches found in the frame with number t. This metric strongly depends on the accuracy of the detector.

The MOTA metric is based on the frequency of false positives, the frequency of missed detections and the frequency of errors in assigning a detection number.

$$MOTA = 1 - \frac{\sum_t (m_t + fp_t + mme_t)}{\sum_t g_t}, \tag{8}$$

where m_t is the number of missed detections, fp_t is the number of false positives, mme_t is the number of mismatches; g_t is the number of objects in the frame t. The index t is responsible for the time or number of the frame.

3 Results and discussion

3.1 GUI

The developed graphic interface (see Fig. 3) is used to navigate through video frames (forward, backward, choice of the frame with a required number), enlarge the window with frames by a given or arbitrary number of times using the mouse, as well as to select and adjust the parameters of the particle tracking algorithm, run the selected tracker with the given parameters, and generate the video in AVI format using the tracker.

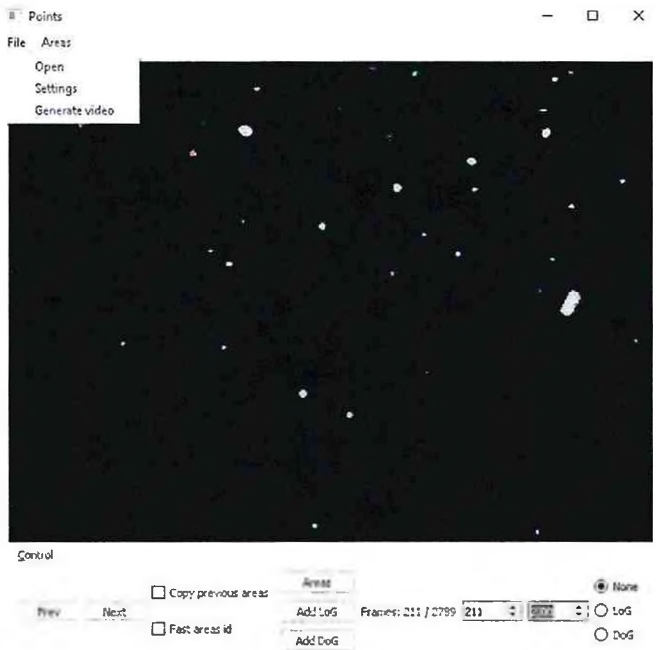

Fig. 3. Graphical user interface.

3.2 Video annotation software

To check the quality of the developed detectors and trackers, there is a need in reference information on the location and trajectory of the flying firebrands. Such information is usually supplied to the input as a special file, in which all detections and tracks of particles are marked and numbered for all frames of the video. Data are usually marked manually. This process is rather long and time-consuming, since a large number of frames have to be marked to test the algorithms. The specialized software accelerates this process by providing the user with quite convenient and effective means for marking data.

For this purpose, an annotation video software was developed and integrated into the main graphical software interface (see Fig. 4). It is used to mark frames in both manual and semi-automatic mode.

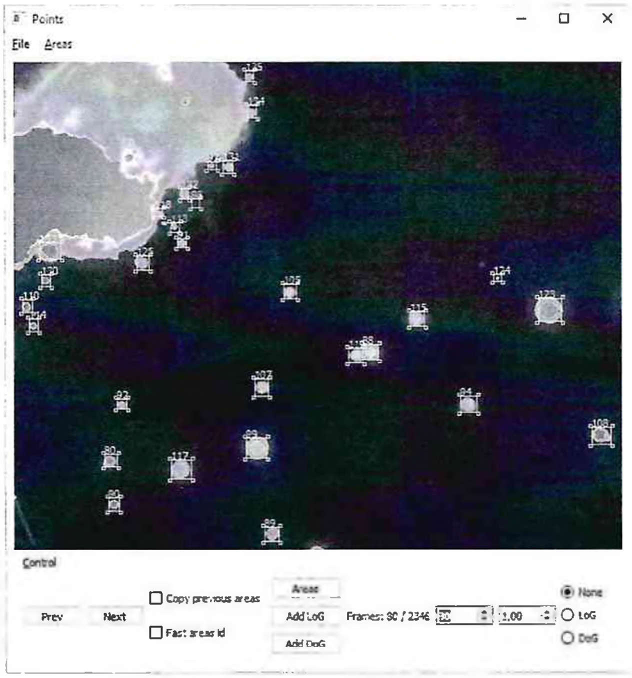

Fig. 4. Frame marked using the annotation video software.

Manual marking is carried out drawing a bounding box around the detection using a mouse. This rectangle has to be assigned a particle number (track number). Rectangles can be moved, copied, deleted and resized.

To assist in annotating video, checkboxes "Copy previous area" and "Fast areas id" are provided in the graphical interface. The first checkbox copies all bounding boxes in the previous frame when moving to the new frame, and the second checkbox automatically numbers the new detections (each new detection will be assigned a number that is larger than the previous number by one).

The interface also includes the "Add LoG" and "Add DoG" buttons which automatically run available particle tracking algorithms (LoG or DoG) in the current frame. If necessary, the operation of automatic detectors can be corrected manually, for example, deleting or adding a new detection, and correcting the number of the particle track. A video can be automatically marked using built-in detectors and trackers.

For quick access to the track by number, there is the "Areas" menu that is used to get or change the coordinates of the bounding box by number. Also, this menu deletes the false detection by its number.

To save and load marked data, a file in the json-format is used [32], in which the frame number, the center coordinates, and the length and width of all bounding boxes marked in it are stored. All detections have unique numbers corresponding to the track number.

Example of the video annotation file:

```
{
   "annotations": [
      {
```

```
        "frame_id": 1,
        "data": [
            {
                "id": "1",
                "bbox": [
                    57,
                    239,
                    22,
                    20
                ]
            },
...
```


The Python3 software language was used to develop the software. To create an application, the PyInstaller program [33] was used, which bundles the Python application and all the necessary dependencies into a single package. Thus, a user can run the software without installing any additional modules in the system.

3.3 Annotated video database

A series of semi-field experiments on thermal mapping of the generation and transport of burning and glowing firebrands were conducted using high-speed infrared cameras. 15 videos of the generation and transport of particles were recorded. Each video is a set of thermograms/frames (see Fig. 5). The time of each experiment ranged from 40 to 80 seconds.

a b

Fig. 5. Example of recorded thermograms/frames (infrared): (a) pine twigs; (b) pellets.

Table with the characteristics of recorded videos is given below.

Table 1. Video database.

File #	Weight of particles, g	Number of particles, pcs	Fuel type	JADE J530SB		FLIR X6530sc	
				Number of frames	Time of recording, s	Number of frames	Time of recording, s
1	---	---	bark/twigs	3125	62	---	---
2	---	---	twigs	3048	61	---	---
3	---	---	twigs	3356	67	---	---
4	---	---	twigs	2789	56	---	---
5	---	---	twigs	2220	44	---	---
6	---	---	twigs	3728	78	---	---
7	---	---	twigs	3978	80	---	---
8	---	---	twigs	3652	7	---	---
9	100	27	pellets	2817	56	2346	48
10	150	127	pellets	2220	44	2346	51
11	150	147	pellets	2789	55	2346	53
12	300	274	pellets	3356	67	2346	61
13	300	266	pellets	3048	60	2346	53
14	300	270	pellets	3411	68	2346	65
15	---	---	twigs	3125	62	2346	57

--- No data available

The video of two experiments was annotated using the developed annotation software. In the future, it is planned to expand the base of annotated videos for verification of the software.

3.4 Detector testing results

The average value of the F1 score (formula 6) in the studied videos was 83% for the DoG algorithm and 73% for the LoG algorithm. The analysis showed that both algorithms can provide the necessary accuracy for the detection of firebrands and are comparable in time and accuracy, but the DoG algorithm is easier controlled and implemented.

In the future work, it is planned to more precisely select the main parameters of the LoG and DoG algorithms, namely the variation range of the Gaussian parameter σ and the threshold value of the local brightness maxima in the image.

3.5 Tracker testing results

The results of testing the quality of trackers are given in Table 2. The average values of all tested videos are indicated as the MOTA and MOTP metrics.

Table 2. Evaluating the quality of tracking algorithms.

Algorithm of tracker and detector	MOTP, %	MOTA,%
Based on the near-	42%	31%

est neighbor, LoG		
Based on the nearest neighbor, DoG	48%	51%
Based on the Hungarian algorithm, LoG	41%	49%
Based on the Hungarian algorithm, DoG	49%	62%

The accuracy of metrics ranges from 0 to 100%, where 100% is absolute coincidence.

The analysis showed the superiority of Hungarian algorithm-based trackers. The MOTA and MOTP metrics have a close or higher accuracy for the Hungarian algorithm. At present, the best result is demonstrated by the method based on the DoG tracking algorithm and the Hungarian algorithm (49% and 62%, respectively). The obtained accuracy is in good agreement with the results of other works. For example, multi object tracking studies [34, 35] show comparable accuracy of work using the applied tracker algorithms.

In the future, it is planned to use a Hungarian algorithm-based tracker to improve the accuracy of its work due to additional selecting the parameters of used metrics and improving the quality of detectors. The accuracy of the detectors used is a comparable value.

4 Conclusion

A series of semi-field experiments was conducted on a unique experimental setup that simulated the transfer of burning and glowing firebrands of pine bark and twigs, as well as wood pellets, depending on the number of particles and different recording parameters using high-resolution infrared cameras. A set of videos was obtained, which was later used for testing and verification of the software developed.

To detect firebrands in the thermograms, the Laplacian of Gaussian (LoG) and the Difference of Gaussian (DoG) algorithms were tested. To evaluate the accuracy of the detectors, an original approach applying the F1 score metric was used. The analysis showed that both metrics can provide the necessary accuracy for the detection of firebrands and are comparable in time and accuracy, but the DoG algorithm is easier controlled and implemented.

Different firebrand tracking algorithms have been developed and tested. In particular, a Hungarian algorithm-based tracker (Kuhn-Munkres algorithm) that tracks the movement of detection between frames with a higher accuracy was implemented. The analysis showed that the Hungarian algorithm-based trackers tracked more precisely the movement of firebrands.

A graphical user interface (GUI) was developed for working with the software and creating an annotated video database. The GUI can be used to perform various manipulations with frames, run a selected detector and a tracker with the specified parameters, as well as to receive the results in the video file.

The further work will be aimed at expanding the database of annotated videos and improving the accuracy of the selected detector and tracker algorithms.

5 Acknowledgements

This work was supported by the Russian Foundation for Basic Research (project #18-07-00548), the Tomsk State University Academic D.I. Mendeleev Fund Program, the Fundamental Research Program of State Academies of Sciences for 2013–2020 (project II.10.3.8) and the Bushfire and Natural Hazard Cooperative Research Centre.

References

1. Mell, W.E., Manzello, S.L., Maranghides, A., Butry, D., Rehm, R.G.: The wildland–urban interface fire problem – current approaches and research needs. International Journal of Wildland Fire **19**, 238–251 (2010). doi: 10.1071/WF07131.
2. Cohen, J.D.: What is the Wildland Fire Threat to Homes? USDA Forest Service Gen. Tech. Rep. PSW-GTR-173. 189–195 (2000).
3. Kornakova, M.; March, A.: Activities in defendable space areas: Reflections on the Wye River-Separation Creek fire. Australian Journal of Emergency Management **32**, 60–66 (2017).
4. Seymat, T.: Deadly wildfires: a devastating year for Portugal. Euronews. (2017).
5. Foote, E I., Manzello, S.L., Liu, J.: Characterizing firebrand exposure during wildland-urban interface fires. In: Proceedings of Fire and Materials 2011 Conference, pp. 479–491. Interscience Communications: San Francisco, CA (2011).
6. Albini, F.A.: Spot fire distance from burning trees- a predictive model. General Technical Report INT-56, USDA Forest Service, Ogden, Utah. (1979).
7. Tarifa, C.S., del Notario, P.P., Moreno, F.G.: On the flight paths and lifetimes of burning particles of wood. In: 10th Symposium (International) on Combustion, 1021–1037. The Combustion Institute (1965).
8. Tse, S.D., Fernandez-Pello, A. and C.: On the flight paths of metal particles and embers generated by power lines in high winds - a potential source of wildland fires. Fire Safety Journal **30**(4), 333–356 (1998).
9. Albini, F.A.: Transport of Firebrands by Line Thermals. Combust. Sci. Technol. **32**(5-6), 277–288 (1983).
10. Manzello, S.L., Park, S., Cleary, T.G.: Investigation on the ability of glowing firebrands deposited within crevices to ignite common building materials. Fire Safety Journal **44**(6), 894–900 (2009). doi:10.1016/j.firesaf.2009.05.001.
11. Suzuki, S., Manzello, S.: Characteristics of Firebrands Collected from Actual Urban Fires. Fire Technology **54**(6), 1533–1546 (2018). https://doi.org/10.1007/s10694-018-0751-x.
12. Filkov, A.; Prohanov, S. Particle Tracking and Detection Software for Firebrands Characterization in Wildland Fires. Fire Technology 1–20 (2018). doi:10.1007/s10694-018-0805-0.

13. Kasymov, D.P., Perminov, V.A., Reyno, V.V., Filkov, A.I., Loboda, E.L.: Experimental setup for producing firebrands to study the spread of wildland fires. Russian Physics Journal **60**(12/2), 107–112 (2017).
14. Loboda, E.L., Kasymov, D.P., Filkov, A.I., Reyno, V.V., Agafontsev, M.V.: Some aspects of field and laboratory research of wildland fires using thermography. In: 30th Intern. Sci.-Pract. Conf. M .: VNIIPO, pp. 295–300 (2018).
15. Kasymov, D.P., Agafontsev, A.M., Filkov, A.I., Perminov, V.V., Reyno, V.V.: Experimental data on the transfer of burning and glowing firebrands and the conditions of ignition of fuel bed. In: the 24th Intern. Symposium, pp. 63–66. Atmospheric and Oceanic Optics. Atmospheric Physics: Proceedings, Tomsk, (2018).
16. Kasymov, D.P., Perminov, V.V., Filkov, A.I., Agafontsev, A.M., Reyno, V.V., Gordeev, E.V.: Generator of burning and glowing firebrands. Patent RF, No. 183063. 2018.
17. El Houssami, M., Mueller, E., Filkov, A. et al.: Experimental procedures characteris-ing firebrand generation in wildland fires. Fire Technology **52**, 731–751 (2016). https://doi.org/ 10.1007/s10694-015-0492-z
18. Filkov, A.I., Prohanov, S.A., Mueller, E., Kasymov, D.P. et al.: Investigation of firebrand production during prescribed fires conducted in a pine forest. Proceedings of the Combustion Institute **36**(2), 3263–3270. (2017).
19. Thomas, J.C., Mueller, E.V., Santamaria, S. et al.: Investigation of firebrand generation from an experimental fire: development of a reliable data collection methodology. Fire Safety Journal **91**, 864–871 (2017). https://doi.org/10.1016/j.firesaf.2017.04.002.
20. Tkinter -Python interface to Tcl/Tk. https://docs.python.org/3/library/tkinter.html, last accessed 2018/12/21.
21. PyGObject. https://pygobject.readthedocs.io, last accessed 2018/12/22.
22. The GUI toolkit for Python. https://wxpython.org/, last accessed 2018/12/22.
23. Python software foundation. https://pypi.org/project/PyQt5/, last accessed 2018/12/23.
24. Tcl Developer Xchange. https://www.tcl.tk/, last accessed 2018/12/23.
25. wxWidgets Cross-Platform GUI Labrary. https://www.wxwidgets.org/, last accessed 2018/01/12.
26. The Qt Company. https://www.qt.io, last accessed 2018/01/12.
27. Qwt User's Guide 6.1.4. https://qwt.sourceforge.io/, last accessed 2018/01/15.
28. Sasaki, Y.: The truth of the F-measure. Teach Tutor mater **1**(5), 1–5 (2007).
29. Knuth, D.: The Art of Computer Programming. Fundamental Algorithms 1(2), (1973).
30. Munkres, J.: Algorithms for the Assignment and Transportation Problems. Journal of the Society for Industrial and Applied Mathematics **5**(1), 32–38 (1957).
31. Bernardin, K., Stiefelhagen, R.: Evaluating multiple object tracking performance: The CLEAR MOT metrics. Eurasip Journal on Image and Video Processing (2008). doi:10.1155/2008/246309.
32. Introducing JSON. https://json.org/, last accessed 2018/01/20.
33. PyInstaller freezes (packages) Python applications into stand-alone executables, under Windows, GNU/Linux, Mac OS X, FreeBSD, Solaris and AIX. http://www.pyinstaller.org, last accessed 2018/01/20.
34. Rahul, M.V., Ambareesh, R., Shobha, G.: Siamese network for underwater multiple object tracking. In: the ACM International Conference, pp. 511–516 (2017).
35. Hua, W., Mu, D., Zheng, Z., Guo, D.: Online multi-person tracking assist by high-performance detection. Journal of Supercomputing, 1–19 (2017).