Supplementary Materials: Orthorectification of Helicopter-Borne High Resolution Experimental Burn Observation from Handheld Imager

Ronan Paugam, Martin J. Wooster, William E. Mell , Mélanie C. Rochoux, Jean-Baptiste Filippi, Gernot Rücker , Olaf Frauenberger, Eckehard Lorenz, Wilfrid Schroeder, Bruce Main and Navashni Govender

Introduction

This document provides descriptions of the image processing algorithms used in the associated manuscript to orthorectify Long Wave Infrared (LWIR), Middle Infrared (MIR) and visible (VIS) handheld helicopter-borne imagery collected during a prescribed burn campaign conducted in savannah type fuel in Kruger National Park in South Africa.

S1. Orthorectification Using Background Image: Algorithms 1

S1.1. Theoretical Basis and Assumption:

This methodology is based on two algorithms: orthorectification with input from a Digital Elevation Model (DEM) and GCPs, and image alignment (i.e., image registration) that computes warping transformation on template image. The latter is at the core of the proposed methodology. It corrects for camera motion, allowing for displacement to be extracted from the image time series.

The limitation of this methodology is that parallax effects caused by the topography or objects in the scene (e.g., trees) need to remain negligible, so that we can neglect computing stereo-rectification, and only consider alignment by projective transformation. This implies that several constraints need to be imposed on the experimental setup. In the scenarios of our experiment, the higher the camera elevation, the lower the parallax effects would be. However, with increasing hovering altitude, salient features would be less resolved, making image registration less efficient. There is, therefore, a trade-off to reach between parallax effects and feature detection. In that sense, we impose the camera line of sight to remain near a mean direction. In other words, the hovering platform is not allowed to spin around the scene. This constraint forces features in the scene to keep the same aspects on the images. In addition to help image registration, images are required to include some background areas, i.e. part of the background scene (as opposed to the fire scene) which is outside the perimeter of the selected plot to be burned. This increases overlap between consecutive images and provides more trustable features (not altered by the fire) that can be tracked over longer time intervals, therefore facilitating alignment with images older than the previous image. A last constraint comes from helicopter operation, for which a lower altitude is better. As air get thinner with altitude, it becomes more difficult to hover, the rotor needs to provide more lift, therefore requiring more power from the engine. This results in (i) increasing gas consumption hence reducing flight time duration, and (ii) engine overheating due to limited airflow during hovering flight mode.

To respect those constraints and in accord with the dimensions of the savanna plots available in KNP (characteristic length of 300 m, see Table 1 in main document), we found for plots smaller than 8 ha that a hovering altitude of around 600 m provided a good balance between feature detection (e.g. bushes, tree), plot size (field of view including the background area around the plot), and helicopter requirement (hovering time much longer than the fire duration).

To simplify further the approach, we assume (*a*) a flat terrain, and (*b*) camera calibration based on pinhole models. The former is justified for the KNP14 data set by the low standard deviation of the DEM departure to respective equivalent flat surface model (see Table 1 in main document). The latter assumes that both LWIR and VIS cameras have good enough optics to be modeled at first order with pinhole model. [1] propose a methodology to calibrate a camera according to the pinhole model with lens distortion correction which

is usually the major limitation of such model. These two assumptions have the advantages of making the computation of orthorectification and image alignment equivalent (I_0^{rec} and I_0 are two planar representations of the same surface), and simpler, as two images of the same planar surface can be related by a homography matrix H [2]. If I_p^m is the image I_p warped on the I_m projection, and X_p is the pixel coordinate on I_p, the projected pixel coordinates X_p^m are expressed with a linear transformation

$$\lambda X_p^m = H_m^p X_p, \tag{S1}$$

where H_m^p is a 3×3 matrix, the pixel coordinates are express in homogeneous space $X_p = (i, j, 1)$, and λ is a scaling factor. In the following, a simplified notation

$$I_p^m \backsimeq H_m^p I_p, \tag{S2}$$

is used for the warping transformation of I_p on the I_m perspective. It computes X_p^m as in Equation (S1) and runs a 2D linear interpolation on the I_m frame.

Two alignment methods are used in our approach: feature- and area- based methods.

Feature-based alignment methods are formed from a detector that select features of interest in the input image, a matching scheme that finds corresponding matches in the template image and a solver that computes the homography matrix in Equation (S1). For the two first steps, the pyramidal implementation of the Lucas-Kanade feature tracker coupled with an optical flow estimation between images is used hereafter PyLkOpt [3]. It is robust against the degree of the geometric deformation, while it may fail when the image content is weakly-textured. PyLkOpt maximizes brightness constancy between a window located around a feature, iterating over image resolution starting at coarse resolution to gradually include image details in the process (a.k.a. the pyramid implementation). The associated solver is the RANSAC method [4], selected for its capability to filter outliers in matching pixels pairs. The specific implementation of PyLkOpt in Algorithms 1 is sketched in Figure S1.

The second area-based method maximize the brightness constancy across the whole image. Here, we used the Enhanced Correlation Coefficient (ECC) algorithm. Using the estimation of H_m^p from PyLkOPt, ECC refines it to maximize the 2D correlation between the images [5]. Considering two images I_p and I_m, with mask M_p and M_m that mask out undesirable pixels, the 2D correlation coefficient that controls image alignment is defined by

$$\rho_{\text{ECC}}(I_p, I_m, M_p, M_m) = \sum_{i \in M_p \cap M_m} \widehat{I_p}(i) \widehat{I_m}(i) \tag{S3}$$

where i scans the intersection of both mask, and $\hat{I}_\bullet = \overline{I_\bullet} / \left\| \overline{I_\bullet} \right\|$ with $\overline{I_\bullet}$ the zeros mean centered vector of $[I_\bullet(j),\ j \in M_\bullet]$. We also introduce here the function $\rho_{\text{ECC}}^{\text{newVal}}$ that computes the new value of ρ_{ECC} associated with an homography transformation H applied to the input image I_p as

$$\rho_{\text{ECC}}^{\text{newVal}}(H, I_p, I_m, M_p, M_m) = \rho_{\text{ECC}}(H_m^p \cdot I_p, I_m, H_m^p \cdot M_p, M_m). \tag{S4}$$

The strength of ECC is its objective function $f(H_m^p)$

$$f(H_m^p) = \rho_{\text{ECC}}^{\text{newVal}}(H_m^p, I_p, I_m, M_p, M_m) \tag{S5}$$

which is approximate by a linear model making it time efficient [5]. However ECC needs a good initial guess of H_m^p when the deformation is strong, to avoid being trapped in local extrema.

The Implementation of ECC in Algorithms 1 is sketched in Figure S2.

It is based on f_{ECC} that is a pyramidal area-based alignment defined on the ECC implementation of [5], see Figure S3.

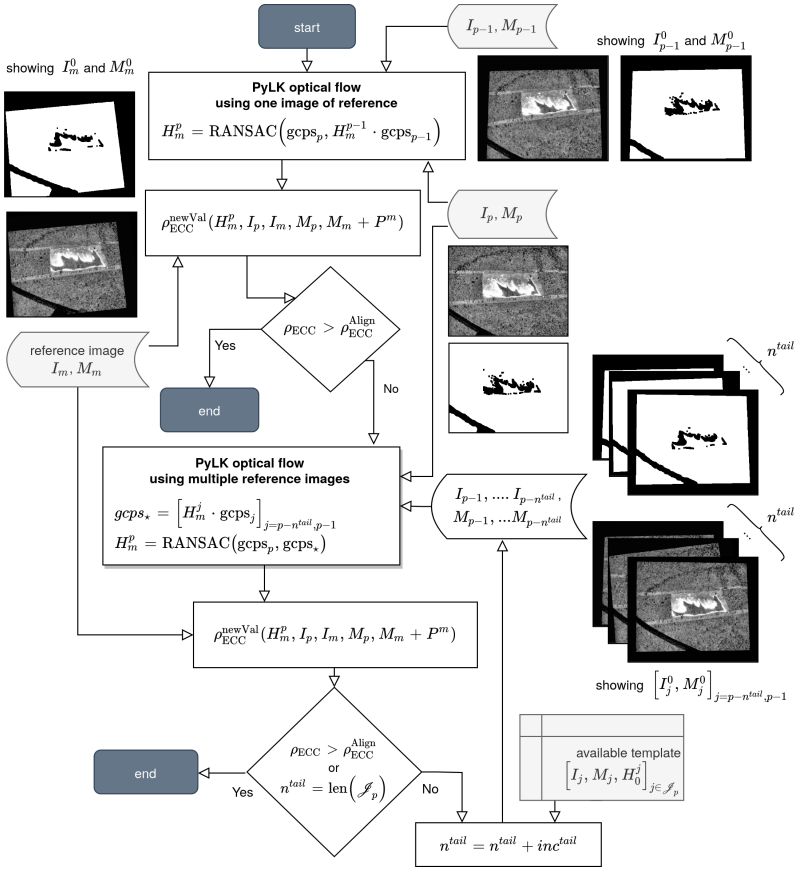

Figure S1. Flowchart representation of the feature-based alignment approach developed for Algorithms 1. Variables and functions are defined in S1.1 and S1.2.

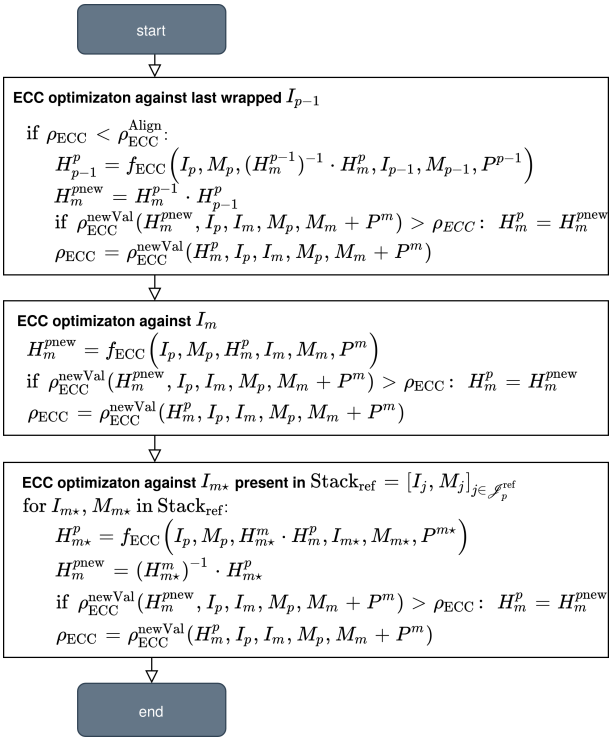

Figure S2. Pseudo code representation of the area-based alignment approach developed for Algorithms 1. Variables are defined in S1.1 and S1.2. f_{ECC} is defined in Figure S3, while $\rho^{\mathrm{newVal}}_{\mathrm{ECC}}$ is defined in Equation (S4).

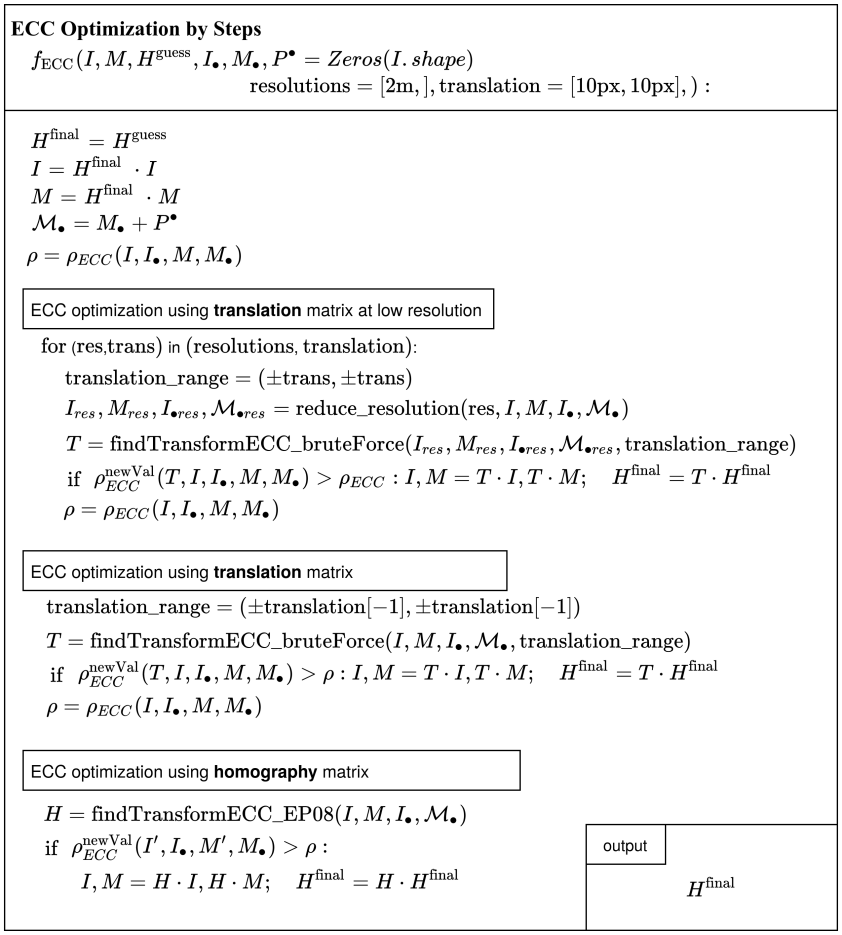

Figure S3. Pseudo code representation of f_{ECC}, a pyramidal area-based alignment algorithm implemented over the Enhanced Correlation Coefficient (ECC) algorithm of [5]. The pyramidal approach is restricted to 2D translation where at each resolution level a simple brute force scan is run over a selected translation range, /ie function findTransformECC_bruteForce in the first two steps. Once the best 2D translation is found, the ECC implementation of [5] (function findTransformECC_EP08 in third step) is run. Variables are defined in S1.1 and S1.2.

S1.2. Algorithms 1 Implementation

The implementation of Algorithms 1 based on iterative image alignment with focus on image background is sketched in Figure S4.

The first step consists in the manual orthorectification of the first image I_0 that is wrapped on a fixed grid centered on the plot with a spatial resolution that is set according to the flight altitude, the sensor resolution, and the field of view of the optics to match an estimate of the native image resolution. Here in our scenario, with a distance between the camera and the plot center slightly higher than 600 m and using specs from the MIR camera, a resolution of $1m$ is chosen, except for the Skukuza4 burn where a resolution of 50 cm is preferred as the plot is smaller and the helicopter is flying at lower altitude.

The implementation of our iterative orthorectification algorithm was developed to work independently of the nature of the image, given that the background scene is seen by the camera sensor. In that sense, the raw inputs (BT^{LWIR} for LWIR or RGB channels for VIS) are converted to 8-bit image raster with histogram equalization [6] coupled with a low pass filter that emphasizes background features for LWIR image (see step (2) in Figure S4 ; $BT < BT^{\text{fire}} = 420\ K$, except for Skukuza4 $BT^{\text{fire}} = 370\ K$). At the same time, each image raster I_p is coupled with a mask M_p that masks out (a) undesirable foreground object (*e.g.* helicopter skid) and (b) hot fire pixel ($BT < BT^{\text{fire}}$). The helicopter skid is masked

using superpixel segmentation [7] and thresholding on gradient magnitude and BT for LWIR, and on gradient direction and 8-bit pixel value for VIS.

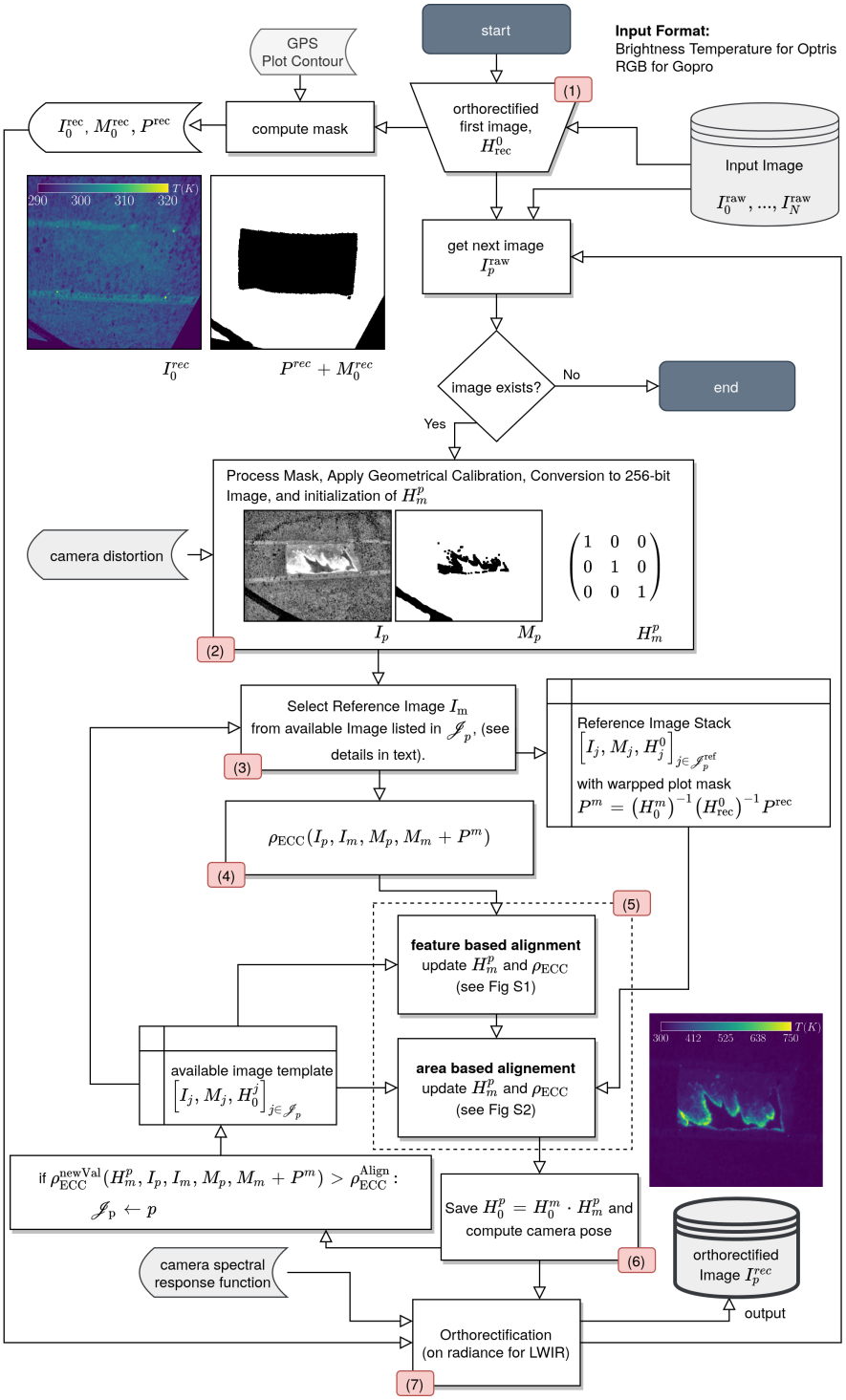

Figure S4. Flowchart representation of Algorithms 1. Variables are defined in S1.1. Numbers in red boxes refer to the algorithm steps discussed in S1.2. Image alignments set in step (5) are further detailed in Figures S1 and S2.

Once I_p and M_p are generated, a reference image is then selected according to the the evolution of ρ_{ECC} time series (step (3) in Figure S4). If, on the last iteration, ρ_{ECC} went below a fixed threshold, $\rho_{ECC}^{\text{Align}}$, for the 4$^{\text{th}}$ consecutive time, then a new reference image,

I_m, is selected within the processed images that are older than a certain time t_{tail} from the current image time. This scheme is set to strengthen the stability of the iterative loop.

Step (4), compute the initial value of $\rho_{ECC}^{newVal}(H_m^p, I_p, M_p, I_m, M_m + P^m)$ with no projection applied to I_p (H_m^p is initialized as the identity matrix). I_p alignment on I_m perspective is then computed in step (5). PyLkOpt is used first to locate matching features between I_p and a stack of n_{tail} warped images. Here, M_p is used to remove fire front pixels in PyLkOpt matching, and using the RANSAC algorithm introduced above, a first guest of the homography H_m^p is estimated. Then comes the area-based optimization (ECC) where H_m^p is optimized against template images I_q selected within previously warped images (see Figure S2 for more details). The new projection transformation is only accepted if it improves ρ_{ECC}, the 2D correlation of I_p and I_m. At this level, the combination of M_p and the plot mask P projected to the template image mask M_q are used to avoid local bias in ECC optimization which has tendency to align close bright pixels together while in presence of non perfectly matching images (*e.g.* due to fire activity). The plot mask P is not used in PyLkOpt when tracking features, at this stage we rely on RANSAC to filter outliers induced by the moving features (*e.g.* front, plume).

ρ_{ECC} is regularly updated in step (5) to control the progress of the alignment. Two thresholds are introduced: ρ_{ECC}^{Ref} is used if an image can be used later as reference image, and $\rho_{ECC}^{\text{Align}}$ that tests if alignment is acceptable. The latter is used to speed up the feature matching alignment (see Figure S1). Experience shows that having $\rho_{ECC}^{\text{Align}} < \rho_{ECC}^{\text{Ref}}$ does not affect the performance of the algorithm, but can greatly improve its completion time.

Once I_p is aligned to the I_m perspective, it can be warped on to the initial image $I_p^0 = H_0^m \cdot H_m^p \cdot I_p$ and orthorectified I_p^{rec} using H_{rec}^0, see last two lines in Algorithms 1 and step (6) and (7) in Figure S4. To easily retrieve the projection transformation between images, I_0 perspective is used a reference, and only the homography matrix H_0^p is saved.

The camera pose, which includes the camera position and orientation in the geographical system, is output at this time of the iterative loop as a diagnostic. For the sake of energy conservation, LWIR images are converted to radiance before application of the transformation $H_0^{\text{map}} \cdot H_m^0 \cdot H_p^m$ that warp it on the fixed grid.

Several parameters were introduced above to control the mechanisms of the algorithm (*i.e.* t_{tail}, n_{tail}, ρ_{ECC}^{Ref}, $\rho_{ECC}^{\text{Align}}$). A sensitivity analysis is proposed in Section 6.1.1 of the main document to estimate them.

S2. LWIR Alignment Optimization based on the Cooling Area: Algorithms 2

S2.1. Implementation

The algorithm presented in this section is designed to optimize the orthorectification transformation estimated in the previous section. Like Algorithms 1, it is designed with an iterative loop (see Algorithms 2) that uses similarity between the current image and the stack of the previous 10 images. It uses a combination of the feature (PyLkOpt) and area (ECC) based alignment described in Figure S5.

The implementation of Algorithm 2 is sketched in Figure S6.

Like in Algorithms 1, a reference image I_m^{rec} is used to compute the 2D correlation coefficient ρ_{ECC} that monitors the stability of the loop (step (1) in Figure S6). I_m is updated whenever the fire alteration of the scene becomes too important. Because the loop is here iterating on already orthorectified images, the rule for updating I_m is modified to make it less aggressive: the new image I_m is the image showing the best ρ_{ECC} within $[I_{p-2\nu^{\text{tail}}}, \ldots, I_{p-\nu^{\text{tail}}}]$ when the moving average of ρ_{ECC} over the last ν^{tail} images is 1% lower than the correlation at the time of I_m.

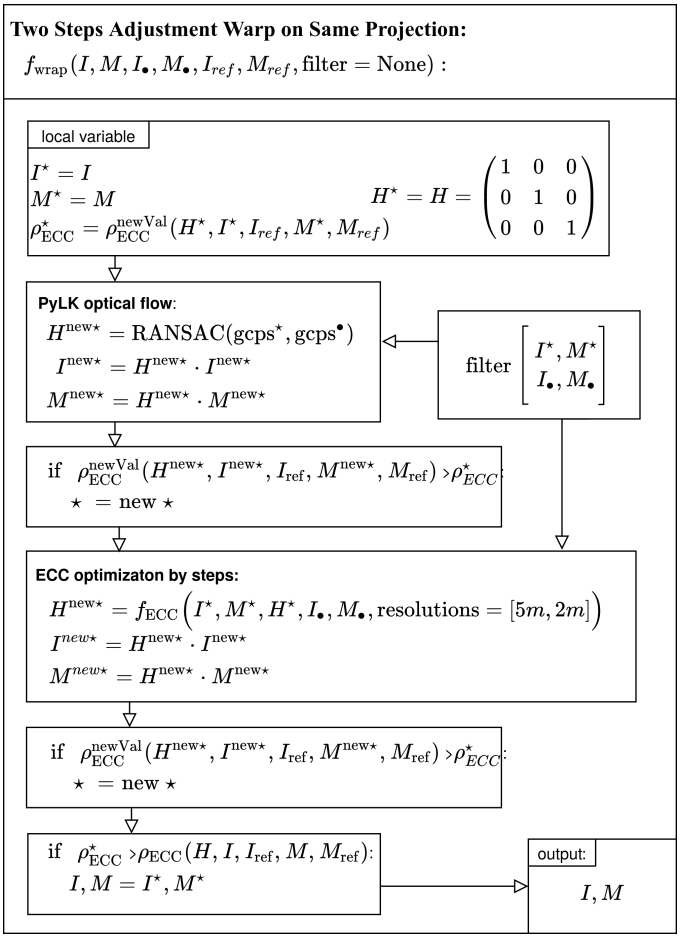

Figure S5. Pseudo code representation of the function f_{warp} used in Algorithms 2 that combined feature and area based alignment onto the same projection reference. Variables are defined in S2.1. The function f_{ECC} is defined in Figure S3.

The main difference between Algorithms 1 and 2 is in the mask computation M_p^{rec}. It is here performed in step (2) as the intersection of (a) the plot mask dilated to include the road that formed the fire break around the plot and (b) the cooling area mask which is formed by the pixels that have been visited by the fire at least a time t^{resi} before the current time t_p of image I_p. This is easily computed from the Arrival Time Map which is updated at the end of each iteration (step 6). Although the time t^{resi} is equivalent to a residence time, that should adapt to the local front ROS and front depth, we choose here to use a constant value for the duration of the fire. This does not maximize the cooling area size, however experience shows that a conservative value of t^{resi} gives good enough improvement of the alignment without having to deal with ROS and flame depth at this stage. Once M_p is computed (note we remove the low pass temperature filter present in Algorithms 1), ρ_{ECC} is first calculated for I_p^{rec} in step (3) using the formulation of Equation (S4) which yields to

$$\rho_{ECC} = \rho^{newVal}_{ECC}(H^p_{adj}, I_p^{\text{rec}}, I_m^{\text{rec}}, M_p^{\text{rec}}, M_m^{\text{rec}}) \tag{S6}$$

where $H^p_{adj} = \nVdash$ for the initialization of step (3). Then ρ_{ECC} is updated to control the advancement of the alignment processes when H^p_{adj} is optimized in steps (4) and (5). These two steps are built on the warping function f_{warp} defined in S2.1 which applies PyLkOpt and ECC successively to align the image I_p^{rec} to a given template image I_q^{rec}. Step (4) calls f_{warp} a single time using masked BT field from image I_p and template image I_{p-1}. Step (5) calls f_{warp} mutiple times using locally normalized BT field from image I_p and template images scanning a stack formed of the 10 previous images and the reference

image I_m. The local normalization filter is defined as an adaptative histogram equalization applied to normalized BT clipped to the temperature range $[300 - 600\ K]$ (see example in Step (5) of Figure S6). It takes as an input parameter the window size l^{LN} that defines the size of the pixels window used in the calculation of the transformation function that performs the local equalization. It enhances low contrast in the cooling area by spreading out the most frequent intensity of the normalized clipped BT distribution. It is important to apply this filter on the orthorectified image. In such cases, all pixels have the same size which makes the adaptative equalization respond better to a temperature distribution resulting from the smoldering structures. Once the adjustment of I_p^{rec} is done, the new orthorectified image is saved and the new state of the fire front is passed to the Arrival Time Map that stored the time advancement of the front. Here are the focus is on the cooling area, a simple threshold temperature ($BT^{LWIR} > 600\ K$) rule is set to determine when the fire reaches a pixel. This choice is largely impacted by flame activity especially in the LWIR spectral range, however experience shows that it gives acceptable cooling area mask estimate when coupled with a conservative value of t^{resi}.

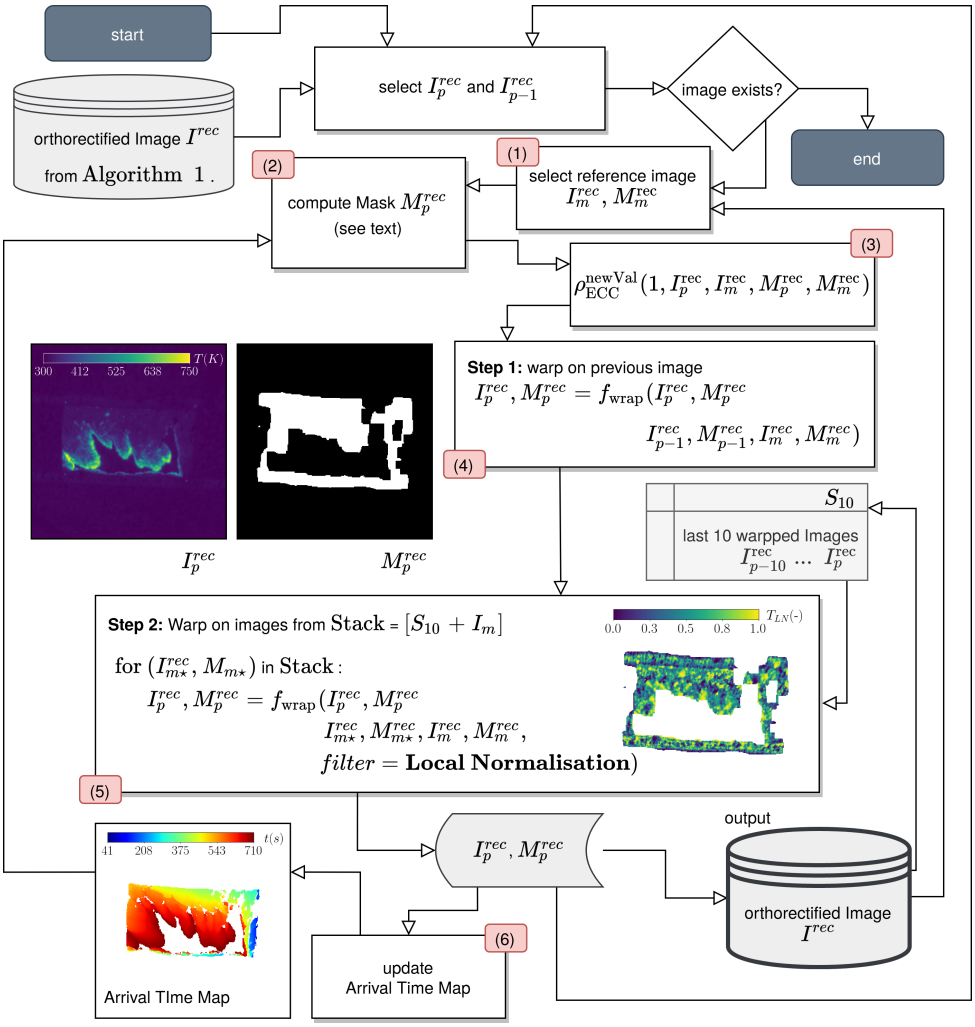

Figure S6. Flowchart representation of Algorithms 2. Variables are defined in S2.1. Numbers in red boxes refer to the algorithm steps discussed in S2.1. f_{warp} is defined in Figure S5. $\rho_{\mathrm{ECC}}^{\mathrm{newVal}}$ is defined in Equation (S4).

S2.2. Parameter Calibration

See Table S1.

Table S1: Parameter values used in Algorithms 2 for the orthorectification of LWIR images for the 4 burns. Parameters are manually evaluated on a fire basis to match the particular fire, *e.g.* average front depth, orthorectification resolution.

parameter	*Skukuza4*	*Skukuza6*	*Shabeni1*	*Shabeni3*	Comments
$T^{\text{thresh}}_{\text{Algo2}}\ (K)$	410	450	420	450	temperature threshold to flag fire pixels in M_p.
$\text{SSIM}^{\text{thresh}}_{\text{Algo2}}\ (-)$	0.45	0.6	0.45	0.5	threshold used in the computation of the steady Mask M_p. Pixel with a mean $\text{SSIM}^{\text{prev}}$ over the last 20 frames are disregarded when they are below $\text{SSIM}^{\text{thresh}}_{\text{Algo2}}$. A lower value helps to keep a larger mask.
$t^{\text{resi}}\ (s)$	30	40	60	30	A residence time marking the depth of the fire front. At time t of the images time series, pixels with fire front arrival time larger than $t - t^{\text{resi}}$ are removed from M_p.
$l^{LN}\ (px)$	60	30	20	40	window size used in the local normalization filter ∇^{LN}: bigger values help to highlight bigger structure or minimize BT noise.

S3. MIR Orthorectification: Algorithms 3–Algorithms 4

S3.1. Algorithms 3 Implementation

Algorithms 3 is designed to align MIR imagery to available concurrent LWIR imagery and used the LWIR orthorectification projection to warp the MIR imagery on the fixed coordinate system grid. Its implementation is sketched in Figure S7.

The is the only algorithm among the four presented in this work that is not recursive. The main loop scans the LWIR images generated with Algorithms 2 as they are available at a lower frame rate than the MIR images. The first 2 steps of an iteration focus on selecting the concurrent MIR image, applying geometrical camera calibration, and setting up a first projective transformation based on the LWIR perspective using the homography matrix derived from MIR and LWIR cameras relative location and orientation on the camera mount. Step (3) computes for both LWIR and MIR the normalized gradient of the locally normalized BT, *i.e.* the ∇^{LN} filter introduced in Section 5.2 of the main document which is designed to emphasize edge contour patterns. The window size l^{LN} used in the local normalization of ∇^{LN} is the only parameter of Algorithms 3. The associated mask to $\nabla^{LN} I^{\text{lwir}}$ and $\nabla^{LN} I^{\text{mir}}$ used to hide unwanted image area from the alignment are also set in step (3). As MIR images only show fire pixels, to keep large areas in both images to run this first alignment only unwanted foreground objects (*i.e.* helicopter skid showing in the LWIR) are masked out. The aim of Algorithms 3 is to provide a good estimation of the MIR/LWIR alignment, to compute a first estimate of MIR orthorectification that is then later optimized in Algorithms 4 while focusing on the cooling area. Therefore, even if the flame front is certainly looking different in near-synchronized MIR and LWIR images, step (4) runs the pyramidal area-based alignment with f_{ECC} set with 3 levels (5, 2, and 1 m, see f_{ECC} in Figure S3). This is assuming that the rear of the front would drive ECC core optimization. The orthorectification of the MIR image is finally completed in step (5) using the combination of the MIR/LWIR alignment and the LWIR orthorectification projection from Algorithms 2.

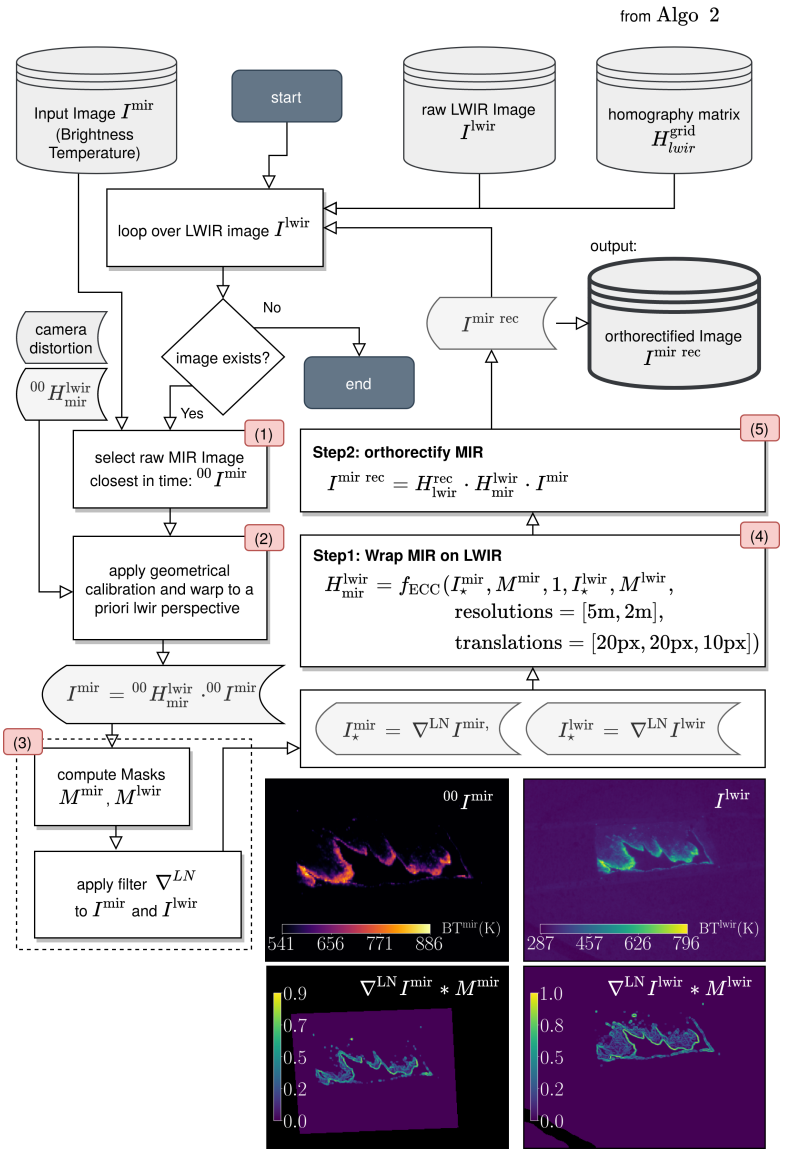

Figure S7. Flowchart representation of Algorithms 3. $^{00}H^{lwir}_{mir}$ is the homography matrix projecting MIR image on LWIR perspective from the pose difference on the camera mount. As cameras are not time synchronized, it is only used as a first guess in step (2). ∇^{LN} is the normalized gradient of locally normalized brightness temperature defined in Section 5.2 of the main document. f_{ECC} is defined in Figure S3. It is a pyramidal area-based alignment set around the ECC implementation of [5].

S3.2. Algorithms 4 Implementation

Algorithms 4 aims to use information from the available orthorectified image of Algorithms 3 to orthorectify the full MIR image time series. A flowchart of its implementation is sketched in Figure S8.

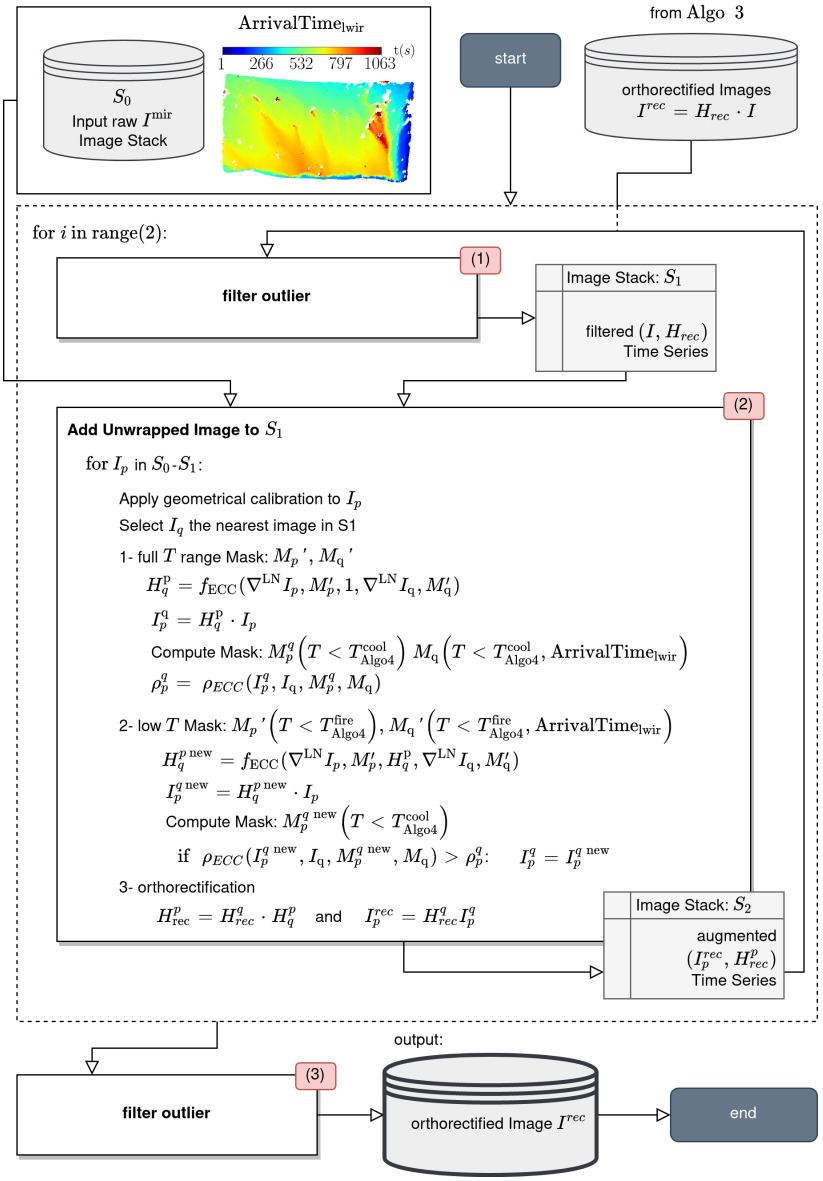

Figure S8. Flowchart representation of Algorithms 4. The algorithm uses as input available orthorectified MIR image from Algorithms 3, and is based on two nested loops. The outer loop runs Filter$^{SSIM}_{MIR}$ (see Section S3.3) and the inner loop scans the unprocessed MIR images to run area-based alignment (see f_{ECC} in Figure S3) between consecutive images. Outlier images flagged in the outer loop are passed to the inner loop as unprocessed images.

The algorithm is design to couple outlier filtering and alignment in a recursive manner. Two nested loops are present. The outer loop runs the image filter to flag poorly orthorectified images, setting them to the unprocessed image stack (see step 1). The inner loop aligns every unprocessed image to its nearest processed image (see step 2). At the end, Algorithms 4 is able to orthorectify most of the MIR image time series, only missing in the worst scenario (*i.e.* Shabeni3 burn) about 10% of the input raw images.

The image to image alignment is set using a pyramidal area-based alignment (see f_{ECC} in Figure S3) that is applied to the filtered image $\nabla^{LN} I_p$ and the nearest available template image $\nabla^{LN} I_q$. f_{ECC} is called two times successively at each iteration of the inner loop, once to align $\nabla^{LN} I$ to $\nabla^{LN} I_{ref}$, using the full image as in Algorithms 3 (empty mask in f_{ECC}), and a second time to align it to the same image but with masks set to only keep the cooling area. For this purpose, the masks M'_p and M'_q are defined (see Figure S8) with a fire temperature threshold T^{fire}_{Algo4} that removes fire pixels. Furthermore, M'_q is complemented

with the projection of the fire front delimitation at time t_q that is captured from the arrival time map computed in Algorithms 2 and projected on the template perspective. As it is applied on raw images, $T^{\text{fire}}_{\text{Algo4}}$ is set according to the hovering altitude. $T^{\text{fire}}_{\text{Algo4}} = 700\ K$ for all burns except for Skukuza4 where $T^{\text{fire}}_{\text{Algo4}} = 650\ K$. Another lower temperature threshold is also used in Algorithms 4 to control the added value of the second call of f_{ECC}. $T^{\text{cool}}_{\text{algo4}} = 600K$ is set as a constant and used to define a new pair of masks (M_p and M_q, see Figure S8) which allows a test of the robustness of alignment improvement.

S3.3. Filter$^{SSIM}_{MIR}$

The time dependent metric ($\text{r}^{\text{low}}_{\text{SSIM}}$) used for the filtering is the ratio of the number of pixel with low $\text{SSIM}^{\text{centre}}_{2D}$ to the total number of pixel present in the cooling area mask M_p of Section 5.2 of the main document. $\text{SSIM}^{\text{centre}}_{2D}$ is defined for a pixel (i, j) as the 20% percentile of $\text{SSIM}(I^{ij}_p, I^{ij}_y)$ with $y \in [p-5 : p+5]$. A Hampel filter is then used to flag image outlier. It is based on a rolling median with a sliding window l^{Hamp} coupled with a standard deviation set according to the median absolute deviation [8]. An image is flagged as an outlier if $\text{r}^{\text{low}}_{\text{SSIM}}$ differs from its local median by more than n^{Hamp} standard deviation. The Filter $\text{Filter}^{\text{SSIM}}_{MIR}$ is made of a recursive loop that takes the full image time series, computes $\text{r}^{\text{low}}_{\text{SSIM}}$, applies the Hampel filter and re-iterates with the new clean image time series until no outliers are present. For all fires, we choose $l^{\text{Hamp}} = 40$ and $n^{\text{Hamp}} = 3$.

References

1. Zhang, Z. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2000**, *22*, 1330–1334. doi:10.1109/34.888718.
2. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*; Cambridge University Press: Cambridge, 2004. doi:10.1017/cbo9780511811685.
3. Bouguet, J.Y. Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm. Technical report, Intel Corporation, Microprocessor Research Labs, 2001. doi:10.1016/j.tim.2005.08.009.
4. Fischler, M.A.; Bolles, R.C. Random sample consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM* **1981**, *24*, 381–395. doi:10.1145/358669.358692.
5. Evangelidis, G.D.; Psarakis, E.Z. Parametric image alignment using enhanced correlation coefficient maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2008**, *30*, 1858–1865. doi:10.1109/TPAMI.2008.113.
6. Zuiderveld, K. Contrast Limited Adaptive Histograph Equalization. Graphic Gems IV; Academic Press Professional: San Diego, 1994; pp. 474–485.
7. Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; Süsstrunk, S. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2012**, *34*, 2274–2281. doi:10.1109/TPAMI.2012.120.
8. Hampel, F.R. The influence curve and its role in robust estimation. *Journal of the American Statistical Association* **1974**, *69*, 383–393. doi:10.1080/01621459.1974.10482962.