SUMARI

Sumari ........................................................................................................................................ 1
A  Pressupost ................................................................................................................................ 3
    A.1  Recursos materials ............................................................................................................ 3
    A.2  Recursos humans ............................................................................................................... 3
    A.3  Import total ....................................................................................................................... 4
B  Avaluació d’impacte ambiental ................................................................................................. 5
    B.1  Descripció general del projecte ......................................................................................... 5
    B.2  Estudi d’alternatives ......................................................................................................... 5
    B.3  Descripció del medi ............................................................................................................ 5
    B.4  Identificació i valoració dels impactes sobre el medi ......................................................... 5
        B.4.1  Criteris d’avaluació .................................................................................................. 5
        B.4.2  Identificació d’impactes. Llista d’activitats ................................................................. 6
        B.4.3  Impactes ambientals potencials ................................................................................. 6
        B.4.4  Valoració dels impactes ............................................................................................. 7
    B.5  Mesures correctores previstes ............................................................................................ 7
C  Algoritmes ................................................................................................................................. 9
    C.1  Cel·les .................................................................................................................................. 9
        C.1.1  analisi_drop_5_zones.m .............................................................................................. 9
        C.1.2  fer_homografies.m ..................................................................................................... 18
        C.1.3  seleccio_isocrones.m .................................................................................................. 21
        C.1.4  seleccio_drop.m ........................................................................................................... 23
        C.1.5  informe_cat.m ............................................................................................................. 24
    C.2  Interfícies ............................................................................................................................ 31
        C.2.1  escriu_punts.m .............................................................................................................. 31
        C.2.2  modifica_homografies.m ............................................................................................. 34
    C.3  Geometria ........................................................................................................................... 40
        C.3.1  centroid.m ..................................................................................................................... 40
        C.3.2  change_ref.m ............................................................................................................... 40
        C.3.3  createCircumference.m ............................................................................................... 40
        C.3.4  createLine_miki ............................................................................................................ 41
A PRESSUPOST

A continuació, es detalla la relació de despeses ocasionades durant la realització d’aquest projecte. Les diferents despeses es troben desglossades en dues categories: el cost dels recursos materials consumits i el cost dels recursos humans.

A.1 Recursos materials

Els recursos materials consumits durant la realització d’aquest projecte estan associats bàsicament al material d’oficina i al consum d’aigua i llum. Cal assenyalar que tota l’activitat ha estat desenvolupada a les instal·lacions del CERTEC (Centre d’Estudis del Risc Tecnològic) i que per tant, s’ha gaudit del material informàtic de què disposa aquesta institució. En conseqüència, en referència a aquest aspecte es considerarà l’amortització dels equips informàtics propietat del CERTEC. Pel que fa al programari emprat, es considera que el cost de les llicències del sistema operatiu (Microsoft® Windows XP Professional) i de la suite ofimàtica (Microsoft® Office 2007) vénen inclosos dins el cost d’adquisició de l’ordinador. En canvi, quant a les llicències del programari específic que s’ha fet servir (Matlab® 7.8 i ThermaCAM Researcher 2001) el cost ha estat nul ja que s’ha gaudit de les llicències de la UPC i del CERTEC.

A la Taula A.1 es presenten aquests costos.

<table>
<thead>
<tr>
<th>Concepte</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amortització dels equips informàtics</td>
<td>160,00 €</td>
</tr>
<tr>
<td>Material d’oficina</td>
<td>73,70 €</td>
</tr>
<tr>
<td>Subministres</td>
<td>70,20 €</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>303,90 €</strong></td>
</tr>
</tbody>
</table>

A.2 Recursos humans

En aquest apartat de costos associats als recursos humans, es contemplen bàsicament les hores de dedicació de la persona encarregada del projecte, així com de les altres dues persones que hi han tingut una participació activa i que han realitzat, respectivament, tasques de direcció i assessorament.

Així, es considera un cost de 45 €·hora\(^{-1}\) per l’enginyer júnior que ha realitzat el projecte, dedicant-hi 40 hores per setmana durant 8 mesos. La direcció ha estat a càrrec d’una doctora enginyera industrial, amb una dedicació de 2 hores setmanals i amb uns honoraris estimats de 75 €·hora\(^{-1}\). Les tasques d’assessorament (d’1 hora setmanal) les ha realitzades una enginyera industrial del CERTEC especialista en la matèria, els honoraris de la qual s’estimen també en 75 €·hora\(^{-1}\).

La Taula A.2, detalla aquests costos.
Desenvolupament d’una metodologia basada en termografia infraroja...

**Taula A.2 Costos associats als recursos humans**

<table>
<thead>
<tr>
<th>Personal</th>
<th>Cost horari [€·hora⁻¹]</th>
<th>Hores dedicades</th>
<th>Cost total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enginyer júnior (projectista)</td>
<td>45</td>
<td>1280</td>
<td>57.600,00 €</td>
</tr>
<tr>
<td>Doctora enginyera industrial (directora)</td>
<td>75</td>
<td>64</td>
<td>4.800,00 €</td>
</tr>
<tr>
<td>Enginyera industrial (assessora)</td>
<td>75</td>
<td>32</td>
<td>2.400,00 €</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td></td>
<td></td>
<td><strong>64.800,00 €</strong></td>
</tr>
</tbody>
</table>

**A.3 Import total**

A continuació es presenta l’import total de les despeses resultants de la realització d’aquest projecte (Taula A.3)

**Taula A.3 Import total associat a la realizació d’aquest projecte**

<table>
<thead>
<tr>
<th>Concepte</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recursos materials</td>
<td>303,90 €</td>
</tr>
<tr>
<td>Recursos humans</td>
<td>64.800,00 €</td>
</tr>
<tr>
<td><strong>SUBTOTAL</strong></td>
<td><strong>65.103,90 €</strong></td>
</tr>
<tr>
<td>Imprevistos (10%)</td>
<td>6.510,39 €</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>71.614,29 €</strong></td>
</tr>
</tbody>
</table>

Així, el cost total de la realització d’aquest projecte és de 71.614,29 €.
B AVALUACIÓ D’IMPACTE AMBIENTAL

B.1 Descripció general del projecte
La realització d’aquest projecte ha consistit bàsicament en el desenvolupament d’una metodologia i una aplicació informàtica capaç d’anàlitzar imatges infraroses d’incendis. Així, l’impacte ambiental d’aquest projecte ve condicionat de manera exclusiva per la realització d’aquestes tasques.

B.2 Estudi d’alternatives
En tractar-se d’un projecte eminentment teòric, es considera que no hi ha solucions alternatives a la seva realització.

B.3 Descripció del medi
Aquest projecte s’ha dut a terme íntegrament a les instal·lacions del CERTEC, ubicat al Departament d’enginyeria química de l’ETSEIB, i ha tingut una durada d’uns 8 mesos. Es considera que la seva realització no ha modificat de manera substancial els nivells de qualitat del medi ambient immediat ni dels elements que el conformen, incloent-hi el context socioeconòmic.

B.4 Identificació i valoració dels impactes sobre el medi
En aquest apartat, s’analitzen els diferents impactes potencials que el projecte podria tenir sobre el medi ambient i que poden ser deguts a:

- l’existència del projecte
- la utilització dels recursos naturals
- l’emissió de contaminants (a l’atmosfera, a l’hidrosfera i a la litosfera), la formació de substàncies nocives o el tractament de residus.

B.4.1 Criteris d’avaluació
A l’Anexo 1 del Real Decreto 1131/1988 de 30 de setembre, pel qual s’aprova el Reglamento para la ejecución del Real Decreto Legislativo 1302/1986, de 28 de Junio, de Evaluación de Impacto Ambiental, s’inclouen un seguit de definicions tècniques imprescindibles per a la valoració dels impactes ambientals considerats significatius. Aquests termes es refereixen als diferents tipus d’efectes que poden donar lloc els impactes derivats d’un projecte i són:

En funció del tipus d’efecte:
- **Efecte positiu.** Aquell que és admès com a tal, tant per la comunitat científica com per la població en general, en el context d’una anàlisi completa dels costos i beneficiis genèrics i de les eventualitats externes de l’actuació contemplada.
- **Efecte negatiu.** Aquell que es tradueix en la pèrdua de valor natural, estètic-cultural paisatgístic, de productivitat ecològica; o en un augment dels perjudicis derivats de la contaminació, de l’erosió i altres riscos ambientals en discordança amb l’estructura ecològica-geogràfica, el caràcter i la personalitat de una localitat determinada.
**En funció de la incidència de l’efecte:**
- **Efecte directe.** Aquell que té una incidència immediata en algun aspecte mediambiental.
- **Efecte indirecte o secundari.** Aquell que suposa una incidència immediata respecte a la interdependència o, en general, respecte a la relació d’un sector ambiental amb un altre.

A continuació, s’indiquen les diferents definicions sobre la magnitud de la valoració d’un impacte ambiental potencial, que fa referència al seu caràcter de compatibilitat ambiental:
- **IA Compatible.** Aquell en el qual la recuperació és immediata en finalitzar l’activitat i no requereix de mesures protectores.
- **IA Moderat.** Aquell en el qual la recuperació no precisa de mesures protectores o correctores intensives, però que per recuperar les condicions inicials requereix un cert temps.
- **IA Sever.** Aquell en el qual la recuperació de les condicions del medi exigeix l’adequació de mesures protectores o correctores, a més d’un període dilatat de temps.
- **IA Critic.** Aquell que presenta una magnitud superior al límit acceptable, produint-se una pèrdua permanent de la qualitat de les condicions ambientals, sense possibilitat de recuperació, ni tan sols amb l’adopció de mesures protectores o correctores. A part d’avaluar els impactes concrets de les diferents relacions causa-efecte, han d’ésser valorats respecte l’impacte global del projecte. La magnitud global del projecte serà positiva si la valoració global és compatible, moderada o severa, mentre que serà negativa si la valoració global és crítica.

### B.4.2 Identificació d’impactes. Llista d’activitats

A l’hora de realitzar una avaluació d’impacte ambiental d’un projecte, hi ha unes fases o activitats principals d’aquest que cal considerar i són la construcció, realització i desmantellament de l’activitat. En aquest cas, però, només té sentit analitzar l’etapa de realització de l’activitat ja que no s’ha de realitzar cap construcció.

Els recursos que han estat necessaris per a desenvolupar aquesta activitat són el consum d’electricitat i el consum de materials fungibles propis d’oficina com paper, cartutxos d’impressora, etc.

### B.4.3 Impactes ambientals potencials

En aquest apartat s’analitzen els possibles impactes ambientals que hagi pogut ocasionar la realització d’aquest projecte. Concretament, es distingeixen dos possibles tipus d’impacte:
- Impacte per l’ús de recursos. Es produeix un impacte directe per l’ús de recursos, bàsicament material d’oficina i energia elèctrica com a conseqüència de l’ús dels equips informàtics, condicionament de l’oficina, etc.
- Impacte per emissions. Igualment es pot considerar l’existència d’un impacte de tipus indirecte derivat del consum d’energia elèctrica que suposa l’emissió de gasos de combustió a les centrals tèrmiques.
B.4.4 Valoració dels impactes
Tots els impactes derivats d’aquest estudi es poden valorar com a impactes compatibles i, per tant, l’impacte global associat al projecte també, ateses les mesures correctores previstes i que es detallen a continuació.

B.5 Mesures correctores previstes
Les mesures correctores previstes per a minimitzar els impactes ambientals se centren bàsicament en la gestió ambiental dels residus generats (paper, cartutxos de tinta d’impressora, etc.).
Desenvolupament d’una metodologia basada en termografia infraroja...
C ALGORITMES

C.1 Cel·les

C.1.1 analisi_drop_5_zones.m

%% SCRIPT D'OBTENCIÓ I ANÀLISI DE DADES (5 ZONES DE COMBUSTIÓ)
%======================================================================
===
% Requeriments previs:
%   * Homografies de les imatges a estudiar mitjançant l'script
contigut
%       al fitxer fer_homografies.m
%   * Màscara del drop obtinguda mitjançant l'script contingut al
fitxer
%       selecció_drop.m
%   * Línies isòcrones amb el temps de captura corresponent
obtingudes
%       amb l'aplicació continguda a l'script seleccio_isocrones.m
% Recomanacions:
%   * Es recomana tenir workspaces desats amb les variables creades
pels
%       scripts fer_homografies.m, selecció_drop.m i
seleccio_isocrones.
%   * Aquests workspaces poden ser importats mitjançant les dues
primeres cèl·lules del present script.

%% Introducció de dades sobre les condicions de descàrrega
%======================================================================
===
clear all;
Va=155; % Direcció de l'avió en graus respecte el Nord (sentit horari)
Vw=291; % Direcció de la qual ve el vent en graus respecte el Nord
(sentit horari)
Vw=Vw-180;
north_phase=180; % Direcció en graus (sentit horari) del nord en el
mapa (0º nord cap amunt);

%% Importació de Ttdi, Ttdf i de la màscara del drop i declaració de
% variables necessàries
%======================================================================
===
dropping_filename='drop_mask_5z10p.mat';
load(dropping_filename,'drop_mask_all','r','c');
T_tditdf_filename='homografies_z2.mat';
load(T_tditdf_filename,'T','Timesec');
T_tditdf=T(:,1:2);
Timesec_tditdf=Timesec(1:2);

%% Importació les homografies corresponents al període de temps que es
vol
Desenvolupament d’una metodologia basada en termografia infraroja...

% estudiar i de les línies isòcrones que marquen el perímetre del foc.
%
======================================================================
===
gps_origin=[485812.45; 6042131.66]; % [Easting; Northing]
signs_map2gps=[-1; 1]; % Xgps=Xorigin+signs_map2gps(1)*Xmap  
% Ygps=Yorigin+signs_map2gps(2)*Ymap

homografies_filename='homografies.mat';
load(homografies_filename,'T','Timesec');

iscochrons_filename='iscochrons.mat';
load(iscochrons_filename,'iscochrons');

ros_filename='ros_vars.mat';
load(ros_filename);

T_time=T;
Timesec_time=Timesec;

z_ini=find(Timesec_time>Timesec_tditdf(2),1);
T=cat(3,T_tditdf,T_time(:,:,z_ini:size(T,3)));
Timesec=cat(1,Timesec_tditdf,Timesec_time(z_ini:size(Timesec,1)));

V=[min(min(T(:,:,2)));360;425;600;700];
mpv=3;
mph=3;
Tamb=303;
%
%% Importació de la màscara del drop al qual està lligat (si s'escau)
%
======================================================================
===
linked_drop_name='Drop 1'; % Escriviu 'none' si no hi ha cap drop lligat
if strcmp(linked_drop_name,'none')==0
    dropping_filename='drop_mask_5z10p.mat';
    T_tditdf_filename='homografies_z2.mat';
    linked_Va=157;
    cp_drop_mask_all=drop_mask_all;
    cp_r=r;
    cp_c=c;
    cp_timesec=Timesec;
    load(dropping_filename,'drop_mask_all','r','c');
    load(T_tditdf_filename,'Timesec');
    linked_drop_mask=drop_mask_all;
    linked_r=r;
    linked_c=c;
    linked_timesec=Timesec(1);
    drop_mask_all=cp_drop_mask_all;
    r=cp_r;
    c=cp_c;
    Timesec=cp_timesec;
end
%
%% Extracció de dades
clear T_drop
T_drop0=NaN(size(drop_mask_all));
for z=1:size(T,3)
    T0=T(:,:,z);
    T_drop0(drop_mask_all)=T0(drop_mask_all);
    T_drop(:,:,z)=T_drop0;
    T_mean(z,6)=mean(T_drop0(drop_mask_all));
    T_min(z,6)=min(min(T_drop0(drop_mask_all)));
    T_max(z,6)=max(max(T_drop0(drop_mask_all)));
    T_std(z,6)=std(T_drop0(drop_mask_all));

    if z>1
        coolingf(z,6)=T_mean(z,6)./T_mean(1,6);
    elseif z==1
        coolingf(z,6)=NaN;
    end
    if z>2
        persistancef(z,6)=Tamb./T_min(z,6);
    elseif z==1||z==2
        persistancef(z,6)=NaN;
    end
% Drop zones over time
[fl,gl,re,bu,ub]=zones5(T_drop0,V);
[zones_distro(z,:),zones_percentage(z,:)]=zones_data5(fl,gl,re,bu,ub);
zones_distro_m2=zones_distro*mpv*mph;

% Creating a mask of the flaming, glowing, residual, burned or unburned drop zones
if z==1
    drop_area=sum(sum(drop_mask_all));
    drop_area_m2=drop_area*mph*mpv;
    T_gradient=(T(:,:,2)-T_drop(:,:,1)).*100./T_drop(:,:,1);
    hotter_px=numel(find(T_gradient>=0));
    uncovered_area=mph.*mpv*hotter_px;
    uncovered_percent=hotter_px./drop_area*100;
    atdif=fl;
    m_atdif=zeros(size(drop_mask_all));
    m_atdif(fl)=1;
    m_atdif=logical(m_atdif);
% Glowing
atdig = gl;
m_atdig = zeros(size(drop_mask_all));
m_atdig(gl) = 1;
m_atdig = logical(m_atdig);

% Residual
atdir = re;
m_atdir = zeros(size(drop_mask_all));
m_atdir(re) = 1;
m_atdir = logical(m_atdir);

% Burned
atdib = bu;
m_atdib = zeros(size(drop_mask_all));
m_atdib(bu) = 1;
m_atdib = logical(m_atdib);

% Unburned
atdiub = ub;
m_atdiub = zeros(size(drop_mask_all));
m_atdiub(ub) = 1;
m_atdiub = logical(m_atdiub);

% Geometry of the drop and its different areas

centroids = [centroid(m_atdif) centroid(m_atdig) centroid(m_atdir) centroid(m_atdib) centroid(m_atdiub) centroid(drop_mask_all)];

  % Distance of each pixel to drop centroid
  size_dist_centroid = [size(drop_mask_all) 3];
  dist_centroid = NaN(size_dist_centroid);
  for i = 1:size(drop_mask_all,2)
      for j = 1:size(drop_mask_all,1)
          if drop_mask_all(j,i) == 1
              dist_centroid(j,i,1) = abs(i-centroids(1,6));
              dist_centroid(j,i,2) = abs(j-centroids(2,6));
          end
      end
  end

  dist_centroid_m = mph * dist_centroid;

  % For the different zones masks:
  dist_centroid_zones(:,:,1) = dist_centroid(:,:,3) .* m_atdif;
  dist_centroid_zones(:,:,2) = dist_centroid(:,:,3) .* m_atdig;
  dist_centroid_zones(:,:,3) = dist_centroid(:,:,3) .* m_atdir;
  dist_centroid_zones(:,:,4) = dist_centroid(:,:,3) .* m_atdib;
  dist_centroid_zones(:,:,5) = dist_centroid(:,:,3) .* m_atdiub;
  dist_centroid_zones(:,:,6) = dist_centroid(:,:,3);
  dist_centroid_zones_m = dist_centroid_zones * mph;
  dist_centroid_zones_m(dist_centroid_zones_m == 0) = NaN;
% figure;
[C,h]=contourf(dist_centroid_zones_m(:,:,6),'LevelStep',4);set(h,'Show Text','on');

10 m to centroid mask

% 10 m to centroid mask

[xcirc,ycirc]=createCircumference(8.5/mph,centroids(1,6),centroids(2,6));
xcirc1=floor(xcirc-0.5)+1;
ycirc1=floor(ycirc-0.5)+1;
circ_mask=poly2mask(xcirc,ycirc,size(T_drop,1),size(T_drop,2));
[rcirc,ccirc]=getedges(xcirc,ycirc);
ttcirc=sub2ind(size(circ_mask),rcirc,ccirc);
circ_mask(ttcirc)=1;

[fl,gl,re,bu,ub]=zones5(T(circ_mask),V);

T_superhot=0.9*max(max(T_drop(:,:,1)));

[xsuperhot,ysuperhot]=ind2sub([size(T_drop,1),size(T_drop,2)],find(T_drop(:,:,1)>

T_superhot(1)));
T_superhot(2)=length(xsuperhot);
T_superhot(3)=sum(sqrt((xsuperhot-

centroids(1,6)).^2+(ysuperhot-

centroids(2,6)).^2)<=10/mph)/T_superhot(2);

hottestcat=find(centroids(1,:)>0,1);
dist_hottestcat=sqrt((centroids(1,6)-

centroids(1,hottestcat))^2+(centroids(2,6)-

centroids(2,hottestcat))^2);

% Search maximum distance and its direction
contorn=[c,r];
[dist_max,extrems,alpha]=dist_max_cont(contorn);
% Change of coordinates to new reference (longitudinal and cross)
[c_corr,r_corr]=change_ref(c,r,alpha(1));
% Plot of the drop in the new reference
contorn_corr=[c_corr-min(c_corr),r_corr-min(r_corr)];
% figure;plot(contorn_corr(:,1),contorn_corr(:,2)); axis image;
% Plot of cross distance along longitudinal direction

[x_dim,y_dim]=mask_dimensions(imrotate(drop_mask_all,alpha(1),'nearest '));
x_dim_m=x_dim.*mph;
y_dim_m=y_dim.*mpv;

% Search maximum distance along flight direction and containing
% drop centroid
flightline=createLine_miki(centroids(1,6),centroids(2,6),tand(90-

north_phase-Va));
dropeges=[c(1:end-1),r(1:end-1),c(2:end),r(2:end)];
```matlab
extremsflight=intersectLineEdge_miki(repmat(flightline,size(dropedges,1),1),dropedges);
    extremsflight=extremsflight(isnan(extremsflight(:,1))==0,:);
    if cosd(90-north_phase-Va)>=0
        alpha_north=270-alpha;
    else
        alpha_north=90-alpha;
    end

    if strcmp(linked_drop_name,'none')==0
        linked_centroid=centroid(linked_drop_mask);
        linked_flightline=createLine_miki(linked_centroid(1),linked_centroid(2),tand(90-north_phase-linked_Va));
        linked_dropedges=[linked_c(1:end-1),linked_r(1:end-1),linked_c(2:end),linked_r(2:end)];
        linked_extremsflight=intersectLineEdge_miki(repmat(linked_flightline,size(linked_dropedges,1),1),linked_dropedges);
        linked_extremsflight=linked_extremsflight(isnan(linked_extremsflight(:,1))==0,:);
        linked_area=sum(sum(linked_drop_mask));
        linked_ros=ros(pos_z_ros(z_linked,2):pos_z_ros(z_tdi,2)-1,:);
        [XI,YI]=meshgrid(1:1:255);
        linked_ros_nonan=linked_ros(isnan(linked_ros(:,7))==0,:);
        [XI,YI,linked_ZI]=griddata(linked_ros_nonan(:,1),linked_ros_nonan(:,2),linked_ros_nonan(:,7),XI,YI);
        linked_ros_field=linked_ZI.*3.*60;
        linked_meanros=nanmean(linked_ros_field(drop_mask_all));
    end

    [fl,gl,re,bu,ub]=zones5(m_atdif.*T_drop0,V);
    [zones_distro_tdif(z,:),zones_percentage_tdif(z,:)]=zones_data5(fl,gl,re,bu,ub);

    [fl,gl,re,bu,ub]=zones5(m_atdig.*T_drop0,V);
```

[zones_distro_tdig(z,:),zones_percentage_tdig(z,:)]=zones_data5(fl,gl,
re,ub,ub);

% Initial residual zone evolution at tdf+t
[fl,gl,ub,ub]=zones5(m_atdir.*T_drop0,V);

[zones_distro_tdir(z,:),zones_percentage_tdir(z,:)]=zones_data5(fl,gl,
re,ub,ub);

% Initial burned zone evolution at tdf+t
[fl,gl,ub,ub]=zones5(m_atdib.*T_drop0,V);

[zones_distro_tdib(z,:),zones_percentage_tdib(z,:)]=zones_data5(fl,gl,
re,ub,ub);

% Initial unburned zone evolution at tdf+t
[fl,gl,ub,ub]=zones5(m_atdiub.*T_drop0,V);

[zones_distro_tdiub(z,:),zones_percentage_tdiub(z,:)]=zones_data5(fl,gl,
re,ub,ub);

% Computing temperatures variables for each zone
%
======================================================================
===
% Initially flamming
if size(atdif,1)~=0
    T_mean(z,1)=mean(T_drop0(atdif));
    T_min(z,1)=min(min(T_drop0(atdif)));
    T_max(z,1)=max(max(T_drop0(atdif)));
    T_std(z,1)=std(T_drop0(atdif));
    coolingf(z,1)=T_mean(z,1)./T_mean(1,1);
    persistancef(z,1)=Tamb./T_min(z,1);
else
    size(atdif,1)==0
    T_mean(z,1)=NaN;
    T_min(z,1)=NaN;
    T_max(z,1)=NaN;
    T_std(z,1)=NaN;
end

% Initially glowing
if size(atdig,1)~=0
    T_mean(z,2)=mean(T_drop0(atdig));
    T_min(z,2)=min(min(T_drop0(atdig)));
    T_max(z,2)=max(max(T_drop0(atdig)));
    T_std(z,2)=std(T_drop0(atdig));
    coolingf(z,2)=T_mean(z,2)./T_mean(1,2);
    persistancef(z,2)=Tamb./T_min(z,2);
else
    size(atdig,1)==0
    T_mean(z,2)=NaN;
    T_min(z,2)=NaN;
end
T_max(z,2)=NaN;
T_std(z,2)=NaN;
end

% Initially residual
if size(atdir,1)~=0
T_mean(z,3)=mean(T_drop0(atdir));
T_min(z,3)=min(min(T_drop0(atdir)));
T_max(z,3)=max(max(T_drop0(atdir)));
T_std(z,3)=std(T_drop0(atdir));
coolingf(z,3)=T_mean(z,3)./T_mean(1,3);
persistencef(z,3)=Tamb./T_min(z,3);
elseif size(atdir,1)==0
T_mean(z,3)=NaN;
T_min(z,3)=NaN;
T_max(z,3)=NaN;
T_std(z,3)=NaN;
end

% Initially burned
if size(atdib,1)~=0
T_mean(z,4)=mean(T_drop0(atdib));
T_min(z,4)=min(min(T_drop0(atdib)));
T_max(z,4)=max(max(T_drop0(atdib)));
T_std(z,4)=std(T_drop0(atdib));
coolingf(z,4)=T_mean(z,4)./T_mean(1,4);
persistencef(z,4)=Tamb./T_min(z,4);
elseif size(atdib,1)==0
T_mean(z,4)=NaN;
T_min(z,4)=NaN;
T_max(z,4)=NaN;
T_std(z,4)=NaN;
end

% Initially unburned
if size(atdiub,1)~=0
T_mean(z,5)=mean(T_drop0(atdiub));
T_min(z,5)=min(min(T_drop0(atdiub)));
T_max(z,5)=max(max(T_drop0(atdiub)));
T_std(z,5)=std(T_drop0(atdib));
coolingf(z,5)=T_mean(z,4)./T_mean(1,4);
persistencef(z,5)=Tamb./T_min(z,4);
elseif size(atdiub,1)==0
T_mean(z,5)=NaN;
T_min(z,5)=NaN;
T_max(z,5)=NaN;
T_std(z,5)=NaN;
end
end

% ROS Grid
% [ros,pos_z_ros]=rateofspread(isochrons);
% [XI,YI]=meshgrid(1:1:255);
% ros_nonan=ros(isnan(ros(:,7))==0,:);
% [XI,YI,ZI]=griddata(ros_nonan(:,1),ros_nonan(:,2),ros_nonan(:,7),XI,YI);
%% ros_field=ZI.*3.*60;
%%
[XI,YI,ros_direction_field]=griddata(ros_nonan(:,1),ros_nonan(:,2),ros_nonan(:,10),XI,YI);
%%
[XI,YI,ros_cosinus]=griddata(ros_nonan(:,1),ros_nonan(:,2),ros_nonan(:,8),XI,YI);
%%
[XI,YI,ros_sinus]=griddata(ros_nonan(:,1),ros_nonan(:,2),ros_nonan(:,9),XI,YI);
ros_drop_direction=atan2(mean(ros_sinus(drop_mask_all)),mean(ros_cosinus(drop_mask_all)))*180/pi;

if ros_drop_direction>=0
    Vros_north=90+north_phase-ros_drop_direction;
else
    Vros_north=90-north_phase-ros_drop_direction;
end
%% imagesc(ros_field); axis xy
C.1.2 *fer_homografies.m*

%
% SCRIPT PER OBTENIR HOMOGRAFIES PER A UN DROP
%
%======================================================================
%==
%%% Carregar les matrius i ordenar-les cronològicament
%
%======================================================================
%==
%clear all
fname=dir('*.mat');
for z=1:size(fname,1)
 load(fname(z).name,'*_DateTime*');
 varname=(strcat(fname(z).name(1:size(fname(z).name,2)-4),
 '_DateTime'));
 DateTime=eval(varname);

Timesec_ini(z,1)=DateTime(1,4)*3600+DateTime(1,5)*60+DateTime(1,6)+DateTime(1,7)/1000;

hora_ini(z,1)=strcat(num2str(DateTime(1,4),'%02d'),':',num2str(DateTime(1,5),'%02d'),':',num2str(DateTime(1,6),'%02d'),',',num2str(DateTime(1,7),'%03d'));
end
[Timesec,order]=sort(Timesec_ini);
for z=1:size(fname,1)
 varname=fname(order(z)).name(1:size(fname(order(z)).name,2)-4);
 load(fname(order(z)).name,varname);
 mat_T(:,:,z)=flipud(eval(varname));
 hora.ordenada{z,1}=hora_ini(order(z),1);
 clear(varname);
end

%% Carregar els punts de referència i el mapa del plot
%
%======================================================================
%==
load('ref_points.mat') % Imports coordinates of the reference points
plot_dim_X=666/3; % X-Dimension of the plot
plot_dim_Y=765/3; % Y-Dimension of the plot

clear mapa_punts_lletres mapa_punts_struct mapa mapa_corr
mapa_punts_lletres={'C1' 'C2' 'C3' 'C4' 'C5' 'B1' 'B2' 'C6' 'C7' 'I2' 'I1' 'C1' 'B3' 'B4' 'C8' 'C7'};
for p=1:size(mapa_punts_lletres,2)
 mapa_punts_struct(1,p)=find(strcmp(refpoints.name,mapa_punts_lletres{p,1}));
end
mapa=[refpoints.xy(mapa_punts_struct,1)
 refpoints.xy(mapa_punts_struct,2)];
mapa_corr=[mapa(:,1) max(mapa(:,2))-mapa(:,2)];
plot(mapa_corr(:,1)/3,mapa_corr(:,2)/3,'O','LineStyle', '-',
 'Color','r','LineWidth',1); axis image; axis xy

% plot_dim_X=max(refpoints.xy(:,1))./3;
% plot_dim_Y=max(refpoints.xy(:,2))./3;

% Si les homografies de Tdi i Tdf ja estan fetes executeu aquesta cel·la
% (en cas contrari passeu directament a la següent)
% Col·loqueu-vos al directori on es troba el workspace que conté les
% homografies i executeu la cel·la
% load('homografies_tdi_tdf.mat','T_tdi','T_tdf')
% T(:,:,1)=T_tdi;
% T(:,:,2)=T_tdf;

%% Homografia

======================================================================

z=1; %Remember to change the z's value when analysing a new image
Tim=400; %If references aren't clearly seen on the IR-image change Tim
valuexy=[B1 B2 C7 C4]; % Change matrix xy according to your own
references on the IR image % B1,B2,C7,C4
mesimatges='Sí';
if z==1
    punts_lletres_ant='Escriviu aquí...';
end
while strcmp('Sí',mesimatges) || fi_modifica==2;

    T0=mat_T(:,:,z);
    plotirconvtemp2(T0,Tim); axis xy; axis image;
    clear punts_struct
    punts_lletres_str=char(escriu_punts(cellstr(punts_lletres_ant)));
    punts_lletres_ant=punts_lletres_str;
    close force
    set(gcf,'Name',['Marqueu els següents punts:
    ,punts_lletres_str],'Number','off')
    n=0;
    p=1;
    while n<size(punts_lletres_str,2)
        n=n+1;
        l=1;
        punt='';
        while n<=size(punts_lletres_str,2) &&
        strcmp(punts_lletres_str(n),',')==0
            punt(l)=punts_lletres_str(n);
            l=l+1;
        n=n+1;
        end
        punts_struct(1,p)=find(strcmp(refpoints.name,punt));
        p=p+1;
    end
    xy=[refpoints.xy(punts_struct,1) refpoints.xy(punts_struct,2)
    ones(size(punts_struct,2),1)]';
    [x,y]=getpts;
    elimina='No';
    while size(x,1)<4 && strcmp(elimina,'No')


Annexos
elimina=questdlg('Voleu suprimir la imatge actual (z=',num2str(z),') d''aquesta seqüència?','Suprimeix la imatge','Sí','No','Sí');
if strcmp(elimina,'Sí')
    [mat_T T,Timesec]=elimina_imatge(mat_T,T,Timesec,z); 
    if le(z,size(mat_T,3))
        waitfor(msgbox_cat(strcat('La imatge ja ha estat suprimida de la seqüència. Ja podeu continuar amb la nova imatge corresponsant a z=',num2str(z),'.'),'Homografia','help'))
    elseif z>size(mat_T,3)
        messagebox_cat('No hi ha més imatges disponibles!','Homografia','warn')
        mesimatges='No';
    end
else
    waitfor(msgbox_cat(strcat('Sisplau, torneu a efectuar la selecció de punts per a realitzar l''homografia de la imatge corresponsant a z=',num2str(z),'.'),'Homografia','help'))
    [x,y]=getpts;
end
if strcmp(elimina,'No')
    uv=[x'; y'; ones(1,size(x,1))];
    H=homography2d(uv,xy);
    % matrius_uv(z,:)=[x' y'];
    matrius_H(z,:)=[H(1,:) H(2,:) H(3,:)];
    close all;
for i=1:plot_dim_X
    for j=1:plot_dim_Y
        uv4 = inv(H)*[3*i;3*j;1];
        uv4 = uv4/uv4(3);
        if any(uv4<[1;1;1])
            T(j,i,z)=NaN;
        elseif any(uv4>[size(T0,2);size(T0,1);1])
            T(j,i,z)=NaN;
        else
            T(j,i,z)=T0(round(uv4(2)),round(uv4(1)));
        end
    end
end
T(:,:,z)=flipud(T(:,:,z));
% figure; contourf(T(:,:,z)); set(gca, 'CLim', [303,800]);
colorbar; axis xy;

% GRÀFIC SOLAPAT

%======================================================================

if z<=size(mat_T,3);
    if z==1
        [fi_modifica,T,Tnova_act]=modifica_homografia([],T(:,:,z),mapa_corr);
    elseif z>1
        [fi_modifica,T]=modifica_homografies(z,T,mapa_corr);
    end
    [fi_modifica,Tnova_ant,Tnova_act]=modifica_homografia(T(:,:,z-1),T(:,:,z),mapa_corr);
end
% if fi_modifica==1
  T(:,:,z)=Tnova_act;
  if z>1
    T(:,:,z-1)=Tnova_ant;
  end
% end
elseif z>size(mat_T,3);
  error=sprintf('Error: valor de z massa elevat. z=%d\n ha de pertànyer al rang z=[2,%d]',z,size(mat_T,3));
  msgbox_cat(error,'Homografia','warn')
end

if z<size(mat_T,3) && fi_modifica==1
  mesimatges=questdlg(strcat('Voleu realitzar l''homografia de la imatge corresponent al següent instant de temps (z=',num2str(z+1),')?'),'Homografia','Sí','No','Sí');
elseif z==size(mat_T,3)
  mesimatges='No';
  msgbox_cat('No hi ha més imatges disponibles!','Homografia','warn')
end
if strcmp('Sí',mesimatges) && fi_modifica==1
  z=z+1;
elseif strcmp('No',mesimatges) && z<size(mat_T,3)
  msgbox_cat(['Per a continuar l''homografia des d''aquest instant de temps preneu z=' num2str(z+1)]','Homografia','help')
end
close all

C.1.3 seleccio_isocrones.m
%% SCRIPT PER A MARCAR LES ISÓCRONES DEL PERÍMETRE D'UN INCENDI
%% Importació de variables necessàries fetes amb fer_homografies.m
%%======================================================================
==
clear all;
homografies_filename='homografies.mat';
load(homografies_filename,'T','Timesec');

%% Selecció del perímetre
z=66; % Primera imatge que es vol marcar
Tm=600;
mesimatges='Sí';

while strcmp('Sí',mesimatges)
  mestrams='Sí';
  numtram=1;
  plotirconvtemp3(T(:,:,z)); axis xy; axis image
  titol_fig=sprintf('Imatge %d de %d, dt:% 0.2f
s',z,size(T,3),Timesec(z)-Timesec(z-1));
  set(gcf,'Name',titol_fig,'Number','off')
  if z>1
    for z_ant=1:z-1
      for numtram_ant=1:isochrons.numsections(z_ant)
Desenvolupament d’una metodologia basada en termografia infraroja...

```matlab
hold on;
plot(isochrons.xy{z_ant,numtram_ant}(:,1),isochrons.xy{z_ant,numtram_ant}(:,2))
end
end
plot(mapa_corr(:,1)./3,mapa_corr(:,2)./3,'O','LineStyle','-','Color','r','LineWidth',1); axis image; axis xy
while strcmp('Sí',mestrams) || strcmp('Repeteix l’últim tram',mestrams)
    if z>1
copiatram_str=sprintf('Voleu copiar el tram %d de la isòcrona anterior?',numtram);
copiatram=questdlg(copiatram_str,'Selecció del perímetre','Sí','No','No');
end
if strcmp('No',copiatram)==1 || z==1
[x,y]=getpts;
formatancada=questdlg('Voleu que el tram seleccionat sigui una forma tancada?','Selecció del perímetre','Sí','No','No');
if strcmp('Sí',formatancada)==1
    x(end+1)=x(1);
y(end+1)=y(1);
end
elseif strcmp('No',copiatram)==0 && z>1
    x=isochrons.xy{z-1,numtram}(:,1);
y=isochrons.xy{z-1,numtram}(:,2);
end
hold on; plot(x,y,'r','LineWidth',2,'Tag','actual');
mestrams=questdlg('Voleu seleccionar més trams?','Selecció del perímetre','Sí','No','Repeteix l’últim tram','Sí');
if strcmp('No',mestrams) || strcmp('Sí',mestrams);
isochrons.numsections{z,1}=numtram;
isochrons.xy{z,numtram}=[x,y];
isochrons.timesec{z,1}=Timesec(z);
end
if strcmp('Sí',mestrams);
    numtram=numtram+1;
    set(findobj('Tag','actual'),'Tag','tram_ok')
elseif strcmp('Repeteix l’últim tram',mestrams)
    set(findobj('Tag','actual'),'Visible','off')
end
end
if z<size(T,3)
mesimatges=questdlg('Voleu realitzar la selecció del perímetre del foc per al següent instant de temps?','Selecció del perímetre','Sí','No','Sí');
if strcmp('Sí',mesimatges)
z=z+1;
elseif strcmp('No',mesimatges)
```
%% SCRIPT DE SELECCIÓ DEL DROP I CREACIÓ DE LA MÀSCARA

%% Requeriments previs:
% * Homografies de les imatges corresponents al Tdi i al Tdf,
%   obtingudes mitjançant l'script contigut al fitxer
%   fer_homografies.m. Aquestes homografies s'han de trobar a
%   T(:,:,1) i T(:,:,2), respectivament.
% Recomanacions:
% * Es recomana tenir un workspace desat amb les variables creades
%   per l'script fer_homografies.m. Aquest workspace pot ser importat
%   mitjançant la primera cèl·lula del present script.

%% Importació d'un workspace
%
clear all
homografies_filename='homografies_z2.mat';
load(homografies_filename,'T');

Tamb=303;
%
T_tdi=T(:,:,1);
T_tdf=T(:,:,2);
Resta_T_noav=T_tdi-T_tdf; % noav significa que no s'ha avançat la
posició de la matriu
Resta_T_noav(find(Resta_T_noav<=0))=NaN; % S'eliminen els punts que
s'han escalfat
Resta_T_noav_p100=Resta_T_noav./T_tdi;% Es calcula el refredament
relatiu a la temperatura inicial
Resta_T_noav_p1000=1000*Resta_T_noav_p100; % Es multiplica per 1000 pq
al fer la representació gràfica es veig millor
%
candidats=Resta_T_noav_p1000.*((T_3390>=V(1)&T_3390<V(2))&Resta_T_noav
_p1000>V_drop(2))|((T_3390>=V(2)&T_3390<V(3))&Resta_T_noav_p1000>V_drop(2))|((T_3390>=V(3)&T_3390<V(4))&Resta_T_noav_p1000>V_drop(2))|((T_3390>=V(4)&T_3390<V(5))&Resta_T_noav_p1000>V_drop(2))|((T_3390>=V(5))&Resta_T_noav_p1000>V_drop(4)); % Matriu Resta_T_noav_p1000 però que conté només els punts candidats a formar part del drop
%
% 2. Selecció de la regió coberta pel drop
Desenvolupament d'una metodologia basada en termografia infraroja...

% V=[min(min(T_tdf));360;425;600;700]; %vector de segmentació:
% [flamming, glowing, residual, burned or preheating or dropped, unburned];
V=[min(min(T_tdf));360;600];
% V_drop=[0;30;100;150];
V_drop=[0;100;150;250;500];
figure; contourf(T_tdi,V,'LineStyle','none'); hold on;
contour(Resta_T_noav_p1000,V_drop,'ShowText','on'); set (gca, 'CLim', [100,800]); axis xy; colorbar %Sobre aquest gràfic es selecciona el contorn del drop
% figure; contourf(T_3390,V); hold on;
contour(candidats,V_drop); %Sobre aquest gràfic es selecciona el contorn del drop

%% (pause per fer zoom)
[x,y]=getpts;
% 3. Creació de la màscara del drop
% 3.1. Arrodonir valors de x,y per a obtenir nombres enters
x1=floor(x-0.5)+1;
y1=floor(y-0.5)+1;
close all
% Aquí ja s'han seleccionat els punts del drop

drop_mask=poly2mask(x1,y1,size(T_tdi,1),size(T_tdi,2));
figure; contourf(T_tdi); hold on; contour(drop_mask); axis xy;
% 3.3. Calcular contorns
[r,c]=getedges(x,y); axis xy;
figure; plot(x,y); axis xy;
figure; plot(c,r); axis xy;
% 3.4. Afeigir contorns a la màscara inicial
drop_mask_all=drop_mask;
tt=sub2ind(size(drop_mask),r,c);
drop_mask_all(tt)=1;
% 4. Crear una nova matriu on només apareguin els valors de temperatura
d'intre del drop
TT=zeros(247,222);
TT(drop_mask_all)=T_tdi(drop_mask_all);
% figure; contourf(TT); axis xy;

C.1.5 informe_cat.m
%% INFORME D'AVALUACIÓ DE DESCÀRREGA AÈRIA
% Aquest informe està orientat a facilitar la resposta de les qüestions
% clau proposades a _Desenvolupament d'una metodologia basada en_
% termografia infraroja per a l'anàlisi quantitativa de productes químics
% en l'extinció d'incendis forestals_.
% warning off
hora=clock;
fprintf('Informe generat el dia %0g/%0g/%0g a les %02g:%02g:%02.0f\n',hora([3,2,1,4,5,6]));
% Detalls de la descàrrega avaluada
experiment_str='AS2E';
descarrega_str='Drop 2';
Annexos

load(strcat(descarrega_str,'_head_drop.mat'),'dx','dy','x_head','y_head');
volum_descarrega=3200; % volum de la descàrrega en litres
% Va=157; % Direcció de l'avió en graus respecte el Nord (sentit horari)
% Vw=291; % Direcció de la qual ve el vent en graus respecte el Nord (sentit horari)
% Vw=Vw-180;
fprintf(1,'Id. Experiment: %s
Id. Descàrrega: %s
',experiment_str,descarrega_str)
%===================================
% Qüestió 1
% La descàrrega ha assolit a l'objectiu desitjat? (Foc puntual, cap, flanc,
% assistència dels mitjans terrestres).
% area(c,r,'FaceColor',[0.71 0.81 0.95],'LineStyle','none','BaseValue',min(r))
hold on; contour(T(:,:,1)); axis image; colormap(ironcolormap)
plot(c,r,'k','LineWidth',2)
% plot(isochrons.xy{z_tdi,1}(:,1),isochrons.xy{z_tdi,1}(:,2),'g')
axis_1=axis;
compassPlot(5,axis_1(2),axis_1(4)-10,north_phase);
xlabel('[pixels]'); ylabel('[pixels]')
if exist('dx','var')==0
    figsize=get(gcf,'Position');
    resizefigures
    f=msgbox_cat('Sisplau, seleccioneu els punts del contorn de la descàrrega que en delimiten la part que ha caigut fora del perímetre del foc. Després premeu la tecla de retorn.','Selecció de d1 i d2','help');
    waitfor(f);
    [dx,dy]=getpts;
    dx=round(dx);
    dy=round(dy);
    f=questdlg('Sisplau, ara seleccioneu els punts del contorn del foc que delimiten la part corresponent al cap. Després premeu la tecla de retorn.','Selecció de x1 i x2','D''acord','No hi ha cap','D''acord');
    waitfor(f);
    if strcmp('D''acord',f)==1
        [x_head,y_head]=getpts;
        x_head=round(x_head);
        y_head=round(y_head);
    else
        x_head=[];
        y_head=[];
    end
%s
save(strcat(descarrega_str,'_head_drop.mat'),'dx','dy','x_head','y_head');
set(gcf,'Position',figsize);
end
if isochrons.numsections{z_tdi}-1==1
    plural='';
else
    plural='s';
end
plural='s';
end

xy_tdi=isochrons.xy{z_tdi,1};
xy_tdi ceil=[ceil(xy_tdi(:,1)) ceil(xy_tdi(:,2))];
dxdy ind=[find2dnearest(dx(1),dy(1),xy_tdi ceil(:,1),xy_tdi ceil(:,2))
 ;find2dnearest(dx(2),dy(2),xy_tdi ceil(:,1),xy_tdi ceil(:,2))];
dx=xy_tdi ceil(dxdy ind(1:2),1);
dy=xy_tdi ceil(dxdy ind(1:2),2);

fprintf('Hi ha %1.f foc%s secundari%s a l''instant
tdi.\n',isochrons.numsections{z_tdi}-1,plural,plural)

% Qüestió 2
% L'objectiu triat era el més adequat? (S'ha fixat com a objectiu la
% zona
% més activa del perímetre de l'incendi o del flank triat?)
% north_phase=180;
color={'r','g','b', 'k', 'y', 'm', 'c'};
z_tdi=sum(cell2mat(isochrons.timesec)<=Timesec(1));
d1=min(dy);
d2=max(dy);
if isempty(x_head)==0
  x_head1=min(y_head);
x_head2=max(y_head);
end
legend_strings={'No cremat','Cremat','Residual','Incandescent','Flama'};
T_isochrone=[];
figure;
for numtram=1:isochrons.numsections{z_tdi}
  xy_tdi=isochrons.xy{z_tdi,numtram};
  xy_tdi ceil=[ceil(xy_tdi(:,1)) ceil(xy_tdi(:,2))];
  T_section=T(sub2ind(size(T),
  xy_tdi ceil(:,2),xy_tdi ceil(:,1),ones(size(xy_tdi ceil,1),1)));
  T_isochrone=[T_isochrone;T_section];
  ros_section=ros_field(sub2ind(size(ros_field),
  xy_tdi ceil(:,2),xy_tdi ceil(:,1),ones(size(xy_tdi ceil,1),1)));
  subplot(2,1,1); hold on;
  plot(mph.*xy_tdi(:,2),T_section,color{numtram})
  subplot(2,1,2); hold on;
  plot(mph.*xy_tdi(:,2),ros_section,color{numtram})
  if numtram==1
    T_stats=[min(T_section) max(T_section)
nanmean(T_section(min(dxdy ind):max(dxdy ind))) nan nan];
    T_stats(4)=(T_stats(3)-T_stats(1))./(T_stats(2)-T_stats(1));
    T_stats(5)=ceil(abs(T_stats(4)/0.25));
    ros_stats=[min(ros_section) max(ros_section)
nanmean(ros_section(min(dxdy ind):max(dxdy ind))) nan nan];
    ros_stats(4)=(ros_stats(3)-ros_stats(1))./(ros_stats(2)-
    ros_stats(1));
    ros_stats(5)=ceil(abs(ros_stats(4)/0.25));
  end
end

% xy_cat=vertcat(isochrons.xy{z_tdi,1:numtram});
subplot(2,1,1); xlabel('Desenvolupament lineal del perímetre [m]');
ylabel('Temperatura aparent [K]');
hold on; annotateplot(2,V(2:end),legend_strings(2:end))
pos_axis=get(gca,'Position');
pos_axis=pos_axis-[0 0 .05 0];
set(gca,'Position',pos_axis);
axis tight
annotateplot(1,mph.*[d1,d2],{'d1','d2'})
if isempty(x_head)==0
    annotateplot(1,mph.*[x_head1,x_head2],{'x1','x2'})
end
axis_2_1=axis;
subplot(2,1,2);
xlabel('Desenvolupament lineal del perímetre [m]');
ylabel('Velocitat de propagació [m/min]');
annotateplot(1,mph.*[d1,d2],{'d1','d2'})
if isempty(x_head)==0
    annotateplot(1,mph.*[x_head1,x_head2],{'x1','x2'})
end
xLim=axis_2_1([1,2]);
pos_axis=get(gca,'Position');
pos_axis=pos_axis
fprintf('Temperatura aparent entre d1 i d2 respecte Tmin i Tmàx de tot el perímetre:
Quartil %0.0g (Percentil %0.0f)
',T_stats(5),T_stats(4)*100)
fprintf('Velocitat de propagació entre d1 i d2 respecte Vp.min i Vp.màx de tot el perímetre:
Quartil %0g (Percentil %0.0f)
',ros_stats(5),ros_stats(4)*100)

%% Qüestió 3
% La descàrrega ha assolit la millor orientació en relació amb el vent, els paramètres de vol i la direcció de propagació del foc?

ajust_plot=[min(contorn(:,1)), min(contorn(:,2))];
figure; plot(contorn(:,1)-ajust_plot(1),contorn(:,2)-ajust_plot(2),k','LineWidth',1.5,'DisplayName','Contorn descàrrega')
hold on; plot([extrems(1,1) extrems(1,3)-ajust_plot(1),[extrems(1,2) extrems(1,4)]-ajust_plot(2),b','DisplayName','Longitud màxima')
plot(extremsflight(:,1)-ajust_plot(1),extremsflight(:,2)-ajust_plot(2),r','DisplayName','Longitud direcció vol');
plot(centroids(1,6)-ajust_plot(1),centroids(2,6)-ajust_plot(2),k','Marker','*','LineStyle','none','DisplayName','Centroide')
axis_compass=axis+[0 5 0 5];
compassPlot(3,axis_compass(2),axis_compass(4),north_phase,[Vw,Va,alpha_north,Vros_north],r,([Vw','Va','Lmax','Vros']);
hold off; axis equal; axis off
legend('Location','SouthOutside'); legend('boxoff');
if cosd(90-north_phase-Va)>=0
    x_dim=fliplr(x_dim);
    x_dim_m=fliplr(x_dim_m);
end
figure; plot(x_dim_m,y_dim_m); axis image
xlabel('Longitud Lmax [m]'); ylabel('Amplada [m]');

% d1d2=sortrows({{find(c==dx(1) & r==dy(1));find(c==dx(2) & r==dy(2))});
% if d1d2(1)==1
Desenvolupament d’una metodologia basada en termografia infrarroja...

```
% cr_p1=[c(1:d1d2(end)),r(1:d1d2(end))];
% cr_p2=[c(d1d2(end):end),r(d1d2(end):end)];
% else
%   cr_p1=[c(d1d2(1):d1d2(end)),r(d1d2(1):d1d2(end))];
%   cr_p2=[c(d1d2(1):d1d2(end)),r(d1d2(1):d1d2(end))];
% end

[Wmax,ind]=max(y_dim_m);
Lmaxrange=round(size(x_dim_m,2)*0.25):round(size(x_dim_m,2))*0.75;
Wmean=mean(y_dim_m(Lmaxrange));
Wstd=std(y_dim_m(Lmaxrange));
fprintf('Àrea de la descàrrega: %1.0f m²

Dosi efectiva: %1.2f l/m²

Perímetre de la descàrrega: %1.0f m

Perímetre exterior: %1.0f m

Perímetre interior: %1.0f m

Amplada mitjana del tram 0.25·Lmax - 0.75·Lmax: %1.1f m

Longitud màxima: %1.0f m

Amplada màxima: %1.0f m (Situada a %1.2f·Lmax)

(Desviació estàndard: %1.2f m)

Longitud L'': %1.0f m

(Direcció %s: %1.fº respecte el nord

V_4_2=V;
figure; [Ccf,hcf]=contourf(T_drop(:,:,1),V_4_2,'DisplayName','Zones de combustió');
set(gca,'CLim', [303, 800])
colorbar('YTickMode','manual','YTick',V_4_2,'YTickLabel',legend_string)
```

%% Qüestió 4
% La descàrrega ha assolit el tram pretès del perímetre? Hi ha zones ja cremades o combustible sense cremar?

figure; area(c,r,'FaceColor',[0.71 0.81 0.95],'LineStyle','none','BaseValue',min(r))
hold on; contour(T(:,:,1)); axis equal; colormap(ironcolormap)
plot(c,r,'k','LineWidth',2)
```
hold on;
contour(T_drop(:,:,1),T_superhot(1),'Fill','on','LineColor','black','LineWidth',2,'DisplayName','0,9<T<Tmàx');
plot(centroids(1,6),centroids(2,6),'k','Marker','*',LineWidth',2,'DisplayName','Centroide')
h=compassPlot(10/mph,centroids(1,6),centroids(2,6));
set(h,'LineStyle',':','DisplayName','Radi de 10 m')
legend('Location','Best');
legend('boxoff')
hold off
pos_axis=get(gca,'Position');
pos_axis=pos_axis-[0 0 .01 0];
set(gca,'Position',pos_axis);
figure; pie(zones_percentage(1,:));
colormap(hot)
legend_strings=fliplr(legend_strings);
legend(legend_strings(find(zones_percentage(1,:)>0)),'Location','NorthEastOutside');
legend('boxoff')
figure; pie(zones_percentage_circ);
colormap(hot)
title('Segmentació a la zona del centroide (R=10 m)')
legend(legend_strings(find(zones_percentage_circ>0)),'Location','BestOutside');
legend('boxoff')
fprintf('Un %0.1f%% dels punts tals que 0.9<T<Tmàx ha caigut en un radi de 10 m del centroide
T_superhot(3)*100)
fprintf('Zona de segmentació de temperatura més alta de la descàrrega: %s
legend_strings(hottestcat))
fprintf('Distància entre el centroide de la descàrrega i el de la zona %s: %.0f m,
legend_strings(hottestcat),mph.*dist_hottestcat)
foradins=('fora','dins');
fprintf('queda, doncs, %s del radi de 10 m del centroide de la descàrrega.
foradins(1+(dist_hottestcat.*mph<=10))
legend_strings=fliplr(legend_strings);

%% Qüestió 5
% Quin percentatge de la longitud de la descàrrega ha resultat més efectiu?
% step_x_Lmax=1*(extrems(3)>extrems(1))-1*(extrems(1)>extrems(3));
x_Lmax=extrems(1):step_x_Lmax:extrems(3);
y_Lmax=tand(alpha).*(x_Lmax-x_Lmax(1))+extrems(2);
step_x_La=1*(extremsflight(2)>extremsflight(1))-1*(extremsflight(1)>extremsflight(2));
x_La=round(extremsflight(1)):step_x_La:round(extremsflight(2));
y_La=tand(90-north_phase-Va).*(x_La-x_La(1))+extremsflight(3);
if cosd(90-north_phase-Va)<=0
    figure; plot(fliplr(x_Lmax-min(x_Lmax)).*mph/abs(cosd(alpha))),T(sub2ind(size(T),round(y_Lmax),x_Lmax,ones(size(x_Lmax)))))
    xlabel('Longitud (direcció Lmax, sentit de vol) [m]');
ylabel('Temperatura apparent [K]');
    hold on; annotateplot(2,V(2:end),legend_strings(2:end))
else
    % Seguint amb el punt anterior
end
Desenvolupament d'una metodologia basada en termografia infraroja...

```matlab
pos_axis=get(gca,'Position');
pos_axis=pos_axis-[0 0 .05 0];
set(gca,'Position',pos_axis);
figure; plot(11(x_La-min(x_La)).*mph/abs(cosd(90-north_phase-Va)),T(sub2ind(size(T),round(y_La),x_La,ones(size(x_La)))))

% xlabel('Longitud (direcció La, sentit de vol) [m]');
ylabel('Temperatura aparent [K]');

if strcmp(linked_drop_name,'none')==1
  fprintf('Aquesta descàrrega no té cap descàrrega precedent. ')
else
  fprintf('Aquesta descàrrega està precedida per la descàrrega %s
precedent.

  sumofmasks=drop_mask_all+linked_drop_mask;
  sumofmasks(sumofmasks==0)=NaN;
  % imagesc(sumofmasks); axis xy
  linked_px=sumofmasks(sumofmasks==2);
  overlapped_area=length(linked_px);

  figure; plot(mph.*c,mpv.*r,'b','DisplayName','Descàrrega actual');
```
hold on;
plot(mph.*linked_c,mpv.*linked_r,'Color','k','LineStyle',':','DisplayName','Descàrrega precedent')
legend('Location','Best')
legend('boxoff')

plot(mph.*extremsflight(:,1),mpv.*extremsflight(:,2),'b','DisplayName','Longitud direcció vol','LineStyle','--');
plot(mph.*centroids(1,6),mpv.*centroids(2,6),'b','Marker','*','LineStyle','none','DisplayName','Centroid')
plot(mph.*linked_extremsflight(:,1),mpv.*linked_extremsflight(:,2),'k','DisplayName','Longitud direcció vol','LineStyle','--');
plot(mph.*linked_centroid(1),mpv.*linked_centroid(2),'k','Marker','*','LineStyle','none','DisplayName','Centroid')
axis equal
hold off
fprintf('Temps (min:seg) entre els tdi de les descàrregues: 
%02g:%02.0f
',floor((Timesec(1)-linked_timesec)/60),mod(Timesec(1)-linked_timesec,60))
fprintf('Àrea solapada: %d m2
',overlapped_area*mph*mpv);
fprintf('L’àrea solapada representa un %01.0f%% de la descàrrega %s
',100*overlapped_area/linked_area,linked_drop_name,100*overlapped_area/descarrega_str)
fprintf('Vpropagació entre els tdi de les descàrregues a la zona %s: %.0f m/min
',descarrega_str,linked_meanros)
end

%% Qüestió 7
% Hi ha zones sense cobrir dins la descàrrega?
% contourf(T_gradient,[10,0,-10,-20,-30,-40,-50,-100]);
set(gca,'CLim',[-60,10]); colormap(jet(7)); colorbar;
axis image; axis off
legend('Gradient de temperatura entre Tdi i Tdf [%]','Location','SouthOutside')
legend('boxoff')
if hotter_px>0
    fprintf('Superfície sense cobrir: %g m2 (un %0.2f % % de la superfície de la descàrrega)\n',uncovered_area,uncovered_percent)
else
    fprintf('La descàrrega ha quedat coberta completament.\n')
end
%
close all
clc

C.2 Interfícies

C.2.1 escriu_punts.m
function varargout = escriu_punts(varargin)
% ESCRIU_PUNTS M-file for escriu_punts.fig
ESCIU_PUNTS, by itself, creates a new ESCIU_PUNTS or raises the existing singleton*. H = ESCIU_PUNTS returns the handle to a new ESCIU_PUNTS or the handle to the existing singleton*. ESCIU_PUNTS('CALLBACK', hObject, eventData, handles, ...) calls the local function named CALLBACK in ESCIU_PUNTS.M with the given input arguments. ESCIU_PUNTS('Property', 'Value', ...) creates a new ESCIU_PUNTS or raises the existing singleton*. Starting from the left, property value pairs are applied to the GUI before escriu_punts_OpeningFcn gets called. An unrecognized property name or invalid value makes property application stop. All inputs are passed to escriu_punts_OpeningFcn via varargin.

"See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one instance to run (singleton)"."

See also: GUIDE, GUIDATA, GUIDATAS

% Last Modified by GUIDE v2.5 23-Jul-2009 19:41:44

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @escriu_punts_OpeningFcn, ...
    'gui_OutputFcn', @escriu_punts_OutputFcn, ...
    'gui_LayoutFcn', [], ... 
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before escriu_punts is made visible.
function escriu_punts_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% varargin command line arguments to escriu_punts (see VARARGIN)

% Choose default command line output for escriu_punts
% handles.output = hObject;
handles.punts_lletres_ant=varargin{1};
guidata(hObject, handles);
set(handles.edit1,'String',handles.punts_lletres_ant)

% UIWAIT makes escriu_punts wait for user response (see UIRESUME)
uiwait;

% --- Outputs from this function are returned to the command line.
function varargout = escriu_punts_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
% varargout{1} = handles.output;
varargout{1} = handles.punts_lletres;

function edit1_Callback(hObject, eventdata, handles)
% hObject handle to edit1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
% str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject handle to edit1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% This function is called when the pushbutton1 is pressed.
% hObject handle to pushbutton1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% If the pushbutton1 is pressed, perform the following actions:
% 1. Clear the contents of the edit1 field.
% 2. Set the background color of the edit1 field to white.
% 3. Update the handles structure with the new data.

% 1. Clear the contents of the edit1 field.
set(hObject,'String','');

% 2. Set the background color of the edit1 field to white.
if ispc
    set(hObject,'BackgroundColor','white');
end

% 3. Update the handles structure with the new data.
handles.punts_lletres_ant = edit1.String;
Desenvolupament d'una metodologia basada en termografia infraroja...

handles.punts_lletres=get(handles.edit1,'String'); % returns contents of edit1 as text
guida(hObject, handles);
uiresume;

% --- Executes when user attempts to close figure1.
function figure1_CloseRequestFcn(hObject, eventdata, handles)
% hObject    handle to figure1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Hint: delete(hObject) closes the figure
delete(hObject);

C.2.2 modifica_homografies.m

function varargout = modifica_homografies(varargin)
% MODIFICA_HOMOGRAFIES M-file for modifica_homografies.fig
% MODIFICA_HOMOGRAFIES, by itself, creates a new
% MODIFICA_HOMOGRAFIES or raises the existing
% singleton*.
% 
% H = MODIFICA_HOMOGRAFIES returns the handle to a new
% MODIFICA_HOMOGRAFIES or the handle to
% the existing singleton*.
% 
% MODIFICA_HOMOGRAFIES('CALLBACK',hObject,eventData,handles,...)
calls the local
% function named CALLBACK in MODIFICA_HOMOGRAFIES.M with the
% given input arguments.
% 
% MODIFICA_HOMOGRAFIES('Property','Value',...) creates a new
% MODIFICA_HOMOGRAFIES or raises the
% existing singleton*. Starting from the left, property value
% pairs are
% applied to the GUI before modifica_homografies_OpeningFcn gets
% called. An
% unrecognized property name or invalid value makes property
% application
% stop. All inputs are passed to modifica_homografies_OpeningFcn
% via varargin.
% 
% "See GUI Options on GUIDE's Tools menu. Choose "GUI allows
% only one
% instance to run (singleton)".
% 
% See also: GUIDE, GUIDATA, GUIHANDLES
% 
% Edit the above text to modify the response to help
% modifica_homografies
% 
% Last Modified by GUIDE v2.5 30-Sep-2009 13:05:41
% 
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
```matlab
% Initialization code

gui_State = struct('gui_Name', mfilename,...
    'gui_Singleton', gui_Singleton,...
    'gui_OpeningFcn', @modifica_homografies_OpeningFcn,...
    'gui_OutputFcn', @modifica_homografies_OutputFcn,...
    'gui_LayoutFcn', [],...'
    'gui_Callback', []);

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% Executing just before modifica_homografies is made visible.
function modifica_homografies_OpeningFcn(hObject, eventdata, handles, varargin)
    % This function has no output args, see OutputFcn.
    % hObject    handle to figure
    % eventdata  reserved - to be defined in a future version of MATLAB
    % handles    structure with handles and user data (see GUIDATA)
    % varargin   command line arguments to modifica_homografies (see VARARGIN)
    handles.scrsz=get(0,'Screensize');
    set(gcf,'Position',[3 34 handles.scrsz(3)-3 handles.scrsz(4)-111]);
    set(gcf,'Position',getnicefigurelocation(get(gcf,'Position'),get(gcf,'Units')));

    handles.seleccio=1;
    handles.vertical_act=0;
    handles.vertical_ant=0;
    handles.horitzontal_act=0;
    handles.horitzontal_ant=0;

    handles.z=varargin{1}; % z
    handles.zant=varargin{1}-1;
    handles.T=varargin{2}; % T
    handles.mapa_corr=varargin{3}; % mapa_corr
    guidata(hObject, handles);

    if handles.z>1
        handles.Tant=handles.T(:,:,handles.z-1);
        handles.Tnova_ant=handles.T(:,:,handles.z-1);
    elseif handles.z==1
        handles.Tant=[];
    end

    handles.Tact=handles.T(:,:,handles.z);
    handles.Tnova_act=handles.T(:,:,handles.z);

    set(handles.edit1,'String',num2str(handles.z));
    guidata(hObject, handles);

    axes(handles.axes1)
```
Desenvolupament d’una metodologia basada en termografia infraroja...

```matlab
contour(handles.Tact,'LineWidth',1.5);axis image; axis xy;
if isempty(handles.Tant)==0;
    hold on; contour(handles.Tant); hold off;
end

axes(handles.axes2)
imagesc(handles.Tact); colormap(ironcolormap); hold on;
plot(handles.mapa_corr(:,1)./3,handles.mapa_corr(:,2)./3,'O','LineStyle','-','Color','r','LineWidth',1); axis image; axis xy
axis off;
uwait;

% UIWAIT makes modifica_homografies wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = modifica_homografies_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1}=handles.fi_modifica;
varargout{2}=handles.T;
if handles.fi_modifica==1||handles.fi_modifica==2
    close all;
end

% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.fi_modifica=1;
guidata(hObject, handles);
uiresume;

% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
handles.fi_modifica=2;
guidata(hObject, handles);
uiresume;

% --- Executes on button press in pushbutton9 (Amunt)
function pushbutton9_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton9 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```
if handles.seleccio==1
    handles.vertical_act=handles.vertical_act+1;

handles.Tnova_act=move_image(handles.Tact,handles.vertical_act,handles.horitzontal_act);
elseif handles.seleccio==2
    handles.vertical_ant=handles.vertical_ant+1;

handles.Tnova_ant=move_image(handles.Tant,handles.vertical_ant,handles.horitzontal_ant);
end
 GUIDATA(hObject, handles);

axes(handles.axes1)
contour(handles.Tnova_act,'LineWidth',1.5);axis image; axis xy;
if isempty(handles.Tant)==0;
    hold on; contour(handles.Tnova_ant); hold off;
end

axes(handles.axes2)
imagesc(handles.Tnova_act); colormap(ironcolormap); hold on;
plot(handles.mapa_corr(:,1)./3,handles.mapa_corr(:,2)./3,'O','LineStyle','-', 'Color','r','LineWidth',1); axis image; axis xy; axis off;

% --- Executes on button press in pushbutton10 (Aval)  
function pushbutton10_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton10 (see GCBO)  
% eventdata   reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
if handles.seleccio==1
    handles.vertical_act=handles.vertical_act-1;

handles.Tnova_act=move_image(handles.Tact,handles.vertical_act,handles.horitzontal_act);
elseif handles.seleccio==2
    handles.vertical_ant=handles.vertical_ant-1;

handles.Tnova_ant=move_image(handles.Tant,handles.vertical_ant,handles.horitzontal_ant);
end
 GUIDATA(hObject, handles);

axes(handles.axes1)
contour(handles.Tnova_act,'LineWidth',1.5);axis image; axis xy;
if isempty(handles.Tant)==0;
    hold on; contour(handles.Tnova_ant); hold off;
end

axes(handles.axes2)
imagesc(handles.Tnova_act); colormap(ironcolormap); hold on;
plot(handles.mapa_corr(:,1)./3,handles.mapa_corr(:,2)./3,'O','LineStyle','-', 'Color','r','LineWidth',1); axis image; axis xy; axis off;

% --- Executes on button press in pushbutton11 (Esquerra)  
function pushbutton11_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton11 (see GCBO)  
% eventdata   reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
if handles.seleccio==1
    handles.horitzontal_act=handles.horitzontal_act-1;

handles.Tnova_act=move_image(handles.Tact,handles.vertical_act,handles.horitzontal_act);
elseif handles.seleccio==2
    handles.horitzontal_ant=handles.horitzontal_ant-1;

handles.Tnova_ant=move_image(handles.Tant,handles.vertical_ant,handles.horitzontal_ant);
end
 GUIDATA(hObject, handles);

axes(handles.axes1)
contour(handles.Tnova_act,'LineWidth',1.5);axis image; axis xy;
if isempty(handles.Tant)==0;
    hold on; contour(handles.Tnova_ant); hold off;
end

axes(handles.axes2)
imagesc(handles.Tnova_act); colormap(ironcolormap); hold on;
plot(handles.mapa_corr(:,1)./3,handles.mapa_corr(:,2)./3,'O','LineStyle','-', 'Color','r','LineWidth',1); axis image; axis xy; axis off;
elseif handles.seleccio==2  
    handles.horitzontal_ant=handles.horitzontal_ant-1;

handles.Tnova_ant=move_image(handles.Tant,handles.vertical_ant,handles.horitzontal_ant);
end
guidata(hObject, handles);
axes(handles.axes1)
contour(handles.Tnova_ant,'LineWidth',1.5);axis image; axis xy;
if isempty(handles.Tant)==0;
    hold on; contour(handles.Tnova_ant); hold off;
end
axes(handles.axes2)
imagesc(handles.Tnova_act); colormap(ironcolormap); hold on;
plot(handles.mapa_corr(:,1)/3,handles.mapa_corr(:,2)/3,'O','LineStyle','-','Color','r','LineWidth',1);
axis image; axis xy; axis off;

% --- Executes on button press in pushbutton12 (Dreta).
function pushbutton12_Callback(hObject, eventdata, handles)
    hObject    handle to pushbutton12 (see GCBO)
    eventdata  reserved - to be defined in a future version of MATLAB
    handles    structure with handles and user data (see GUIDATA)
if handles.seleccio==1
    handles.horitzontal_act=handles.horitzontal_act+1;
handles.Tnova_act=move_image(handles.Tact,handles.vertical_act,handles.horitzontal_act);
elseif handles.seleccio==2
    handles.horitzontal_ant=handles.horitzontal_ant+1;
handles.Tnova_ant=move_image(handles.Tant,handles.vertical_ant,handles.horitzontal_ant);
end
guidata(hObject, handles);
axes(handles.axes1)
contour(handles.Tnova_act,'LineWidth',1.5);axis image; axis xy;
if isempty(handles.Tnova_act)==0;
    hold on; contour(handles.Tnova_ant); hold off;
end
axes(handles.axes2)
imagesc(handles.Tnova_act); colormap(ironcolormap); hold on;
plot(handles.mapa_corr(:,1)/3,handles.mapa_corr(:,2)/3,'O','LineStyle','-','Color','r','LineWidth',1);
axis image; axis xy; axis off;

% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
    hObject    handle to popupmenu1 (see GCBO)
    eventdata  reserved - to be defined in a future version of MATLAB
    handles    structure with handles and user data (see GUIDATA)
    % Hints: contents = get(hObject,'String') returns popupmenu1 contents
    % as cell array
    handles.seleccio=get(hObject,'Value'); % returns selected item from popupmenu1
    if handles.seleccio==1
        set(handles.edit1,'String',num2str(handles.z));
    elseif handles.seleccio==2 && handles.z>1
        set(handles.edit1,'String',num2str(handles.z-1));
end

guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
    hObject    handle to popupmenu1 (see GCBO)
    eventdata  reserved - to be defined in a future version of MATLAB
    handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit1_Callback(hObject, eventdata, handles)
    hObject    handle to edit1 (see GCBO)
    eventdata  reserved - to be defined in a future version of MATLAB
    handles    structure with handles and user data (see GUIDATA)

    % get(hObject,'String'); % returns contents of edit1 as text
    handles.zant=str2double(get(hObject,'String')); % returns contents of edit1 as a double
    handles.Tant=handles.T(:,:,handles.zant);
    handles.horitzontal_ant=0;
    handles.Tant=handles.T(:,:,handles.zant);
    handles.Tnova_ant=handles.T(:,:,handles.zant);
    guidata(hObject, handles);

    axes(handles.axes1)
    contour(handles.Tnova_act,'LineWidth',1.5);axis image; axis xy;
    if isempty(handles.Tant)==0;
        hold on; contour(handles.Tnova_ant); hold off;
    end
    guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
    hObject    handle to edit1 (see GCBO)
    eventdata  reserved - to be defined in a future version of MATLAB
    handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
    get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
% --- Executes on button press in pushbutton13.
function pushbutton13_Callback(hObject, eventdata, handles) % Desa el canvi actual
    % hObject    handle to pushbutton13 (see GCBO)
    % eventdata  reserved - to be defined in a future version of MATLAB
    % handles    structure with handles and user data (see GUIDATA)
    handles.T(:,:,handles.z)=handles.Tnova_act;
    handles.T(:,:,handles.zant)=handles.Tnova_ant;
    guidata(hObject, handles);

% --- Executes on key press with focus on edit1 and none of its controls.
function edit1_KeyPressFcn(hObject, eventdata, handles)
    % hObject    handle to edit1 (see GCBO)
    % eventdata  structure with the following fields (see UICONTROL)
    % Key: name of the key that was pressed, in lower case
    % Character: character interpretation of the key(s) that was pressed
    % Modifier: name(s) of the modifier key(s) (i.e., control, shift) pressed
    % handles    structure with handles and user data (see GUIDATA)

C.3 Geometria

C.3.1 centroid.m
function XY_G=centroid(mask,area)
    if nargin==1
        area=sum(sum(mask));
    end
    % x del centroide
    OX_mask=find(sum(mask,1)>0);
    h_X=sum(mask,1);
    S_OX_drop=sum(OX_mask.*h_X(OX_mask));
    OX_G=S_OX_drop/area;
    % f i de la x del centroide
    % y del centroide
    OY_mask=find(sum(mask,2)>0);
    w_Y=sum(mask,2);
    S_OY_mask=sum(OY_mask.*w_Y(OY_mask));
    OY_G=S_OY_mask/area;
    % f i de la y del centroide
    XY_G=[OX_G;OY_G];

C.3.2 change_ref.m
function [x_newref,y_newref]=change_ref(x,y,alpha)
x_newref=x.*cosd(alpha)+y.*sind(alpha);
y_newref=-x.*sind(alpha)+y.*cosd(alpha);
function [xcirc,ycirc,varargout]=createCircumference(R, X0, Y0,N,plotit)

C.3.3 createCircumference.m
% createCircumference
% createCircumference(R, X0, Y0,N)
% plots a compass indicating north and any desired directions.
% R: radius of the circumference
% X0: x-coordinate of circumference centre
% Y0: y-coordinate of circumference centre
% N: number of (xcirc,ycirc) points created
% plotit: 1-value plots the created circumference
% by Miquel Cubells i Gonzalvo 08/10/2009

% Define and draw the circumference
if nargin==3
    N=100;
end

angle=linspace(0,2*pi-(1/N),N);
xcirc=cos(angle)*R+X0;
ycirc=sin(angle)*R+Y0;

if nargin==5 && plotit==1
    h=plot(xcirc,ycirc, '-k', 'MarkerSize', 2);
    axis equal
    set(get(get(h,'Annotation'),'LegendInformation'),'IconDisplayStyle','off'); % Exclude line from legend
    varargout{1}=h;
end

C.3.4 createLine_miki
function line=createLine_miki(x,y,m)
% createnormalLine creates a line represented in parametric form:
% [x y dx dy], given a point and the slope of the direction vector.
%   x = x0 + t*dx
%   y = y0 + t*dy;

line=[x y cos(atan(m)) sin(atan(m))];

C.3.5 dist_max_cont.m
function [dist_max,extrems,alpha]=dist_max_cont(contorn)

dist_contorn=zeros(size(contorn,1));
num_max=0;
dist_max=0;
alpha=0;

for i=1:size(contorn,1)
    for j=(i+1):size(contorn,1)
        dist_contorn(i,j)=sqrt((contorn(i,1)-contorn(j,1))^2+(contorn(i,2)-contorn(j,2))^2);
        dist_contorn(j,i)=dist_contorn(i,j);
        if dist_contorn(i,j)>dist_max
            dist_max=dist_contorn(i,j);
            num_max=1; % número de distàncies màximes iguals trobades
            extrems=[contorn(i,:) contorn(j,:)];
            cos_alpha=abs(extrems(1,3)-extrems(1,1))/dist_max;
        end
    end
end
Desenvolupament d’una metodologia basada en termografia infraroja...

```matlab
if ((extrems(1,3)>=extrems(1,1))&&(extrems(1,4)>=extrems(1,2)))||(extrems(1,3)<=extrems(1,1))&&(extrems(1,4)<=extrems(1,2))
    signe=1;
else
    signe=-1;
end
alpha=signe*acosd(cos_alpha);
elseif dist_contorn(i,j)==dist_max && dist_contorn(i,j)>0
if [contorn(i,:) contorn(j,:)]~=extrems(num_max,:)
    num_max=num_max+1; % número de distàncies màximes iguals trobades
    extrems(num_max,:)=contorn(i,:); extrems(num_max,1)=dist_contorn(i,j);
    cos_alpha=abs(extrems(num_max,3)-
end
end
end
end
ajust_plot=[min(contorn(:,1)), min(contorn(:,2))];
```

**C.3.6 find2dnearest.m**

```matlab
function ind=find2dnearest(x0,y0,xvec,yvec,num)
% find2dnearest(x0,y0,xvec,yvec,num) returns the index vector of the 
% (xvec,yvec) points closest to (x0,y0).
% x0: x-coordinate of the reference point
% y0: y-coordinate of the reference point
% xvec: vector of x-coordinates
% yvec: vector of y-coordinates
% num: number of indexes to be given (default: 1)

dist_unsorted=hypot(xvec-repmat(x0,length(xvec),1),yvec-
end
ind=all_ind(1:num);
```
C.3.7 interpoolEdge.m

function [r,c,varargout]=interpoolEdge(x,y,N)
% interpoolEdge creates linearly spaced knots between the extremes of an
% edge and interpools them using linear splines.

transpose=0;
if size(x,2)<size(x,1)
    x=x';
    y=y';
    transpose=1;
end

% Prepare x and y
xx0=reshape(repmat(x(1:end-1),N,1),1,N*(length(x)-1));
yy0=reshape(repmat(y(1:end-1),N,1),1,N*(length(y)-1));
xx1=reshape(repmat(x(2:end),N,1),1,N*(length(x)-1));
yy1=reshape(repmat(y(2:end),N,1),1,N*(length(y)-1));

% Create linearly spaced x vector between edges
dx=xx1-xx0;
seq=repmat(0:N-1,1,length(x)-1);
r=[xx0+1/N.*dx.*seq x(end)];

% Calculate slopes for each edge
m=(yy1-yy0)./(xx1-xx0);
m(end+1)=m(end);
% Slope in extreme knots is the average between the slopes of the two
% edges containing it.
m=0.5.*(m+[m(1),m(1:end-1)]);

% Calculate y values for each point
c=[yy0+m(1:end-1).*(r(1:end-1)-xx0) y(end)];

if transpose==1
    r=r';
    c=c';
    m=m';
end
if nargout==3
    varargout{1}=m;
end

C.3.8 intersectLineEdge_miki.m

function point=intersectLineEdge_miki(line, edge)
%INTERSECTLINEEDGE return intersection between a line and an edge
%
P = intersectLine(LINE, EDGE) returns the intersection point of
% lines LINE and edge EDGE. LINE is a 1x4 array containing
% parametric representation of the line ([x0 y0 dx dy], see createLine for
details).
% EDGES is a 1x4 array containing corodinante of first point and
coordinate of second point.

In case of colinear line and edge, returns [Inf Inf].
If line does not intersect edge, returns [NaN NaN].

If each input is [N*4] array, the result is a [N*2] array containing
intersections of each couple of lines.

---------

author : David Legland
INRA - TPV URPOI - BIA IMASTE
created the 31/10/2003.

adapted by: Miquel Cubells i Gonzalvo 5/09/2009

x0 = line(:,1);
y0 = line(:,2);
dx1 = line(:,3);
dy1 = line(:,4);
x1 = edge(:,1);
y1 = edge(:,2);
x2 = edge(:,3);
y2 = edge(:,4);
dx2 = x2-x1;
dy2 = y2-y1;

N1 = length(x0);
N2 = length(x1);

% indices of parallel lines
par = abs(dx1.*dy2-dx2.*dy1)<1e-14;

% indices of colinear lines
col = abs((x1-x0).*dy1-(y1-y0).*dx1)<1e-14 & par;

xi(col) = Inf;
iy(col) = Inf;
x(i) = par & ~col = NaN;
y(i) = par & ~col = NaN;
i = ~par;

% compute intersection points
if N1==N2
    xi(i) = ((y1(i)-y0(i)).*dx1(i).*dx2(i) + x0(i).*dy1(i).*dx2(i) - x1(i).*dy2(i).*dx1(i)) ./ ...
            (dx2(i).*dy1(i)-dx1(i).*dy2(i)) ;
y(i) = ((x1(i)-x0(i)).*dy1(i).*dy2(i) + y0(i).*dx1(i).*dy2(i) - y1(i).*dx2(i).*dy1(i)) ./ ...
            (dx1(i).*dy2(i)-dx2(i).*dy1(i)) ;
elseif N1==1
    xi(i) = ((y1(i)-y0).*dx1.*dx2(i) + x0.*dy1.*dx2(i) - x1(i).*dy2(i).*dx1(i)) ./ ...
            (dx2(i).*dy1-dx1.*dy2(i)) ;
yi(i) = ((x1(i)-x0).*dy1.*dy2(i) + y0.*dx1.*dy2(i) -
y1(i).*dx2(i).*dy1) ./ ...
   (dx1.*dy2(i)-dx2(i).*dy1) ;
elseif N2==1
    xi(i) = ((y1-y0(i)).*dx1(i).*dx2 + x0(i).*dy1(i).*dx2 -
x1(i).*dy2.*dx1(i) ) ./ ...
   (dx2.*dy1(i)-dx1(i).*dy2) ;
yi(i) = ((x1-x0(i)).*dy1(i).*dy2 + y0(i).*dx1(i).*dy2 -
y1(i).*dx2.*dy1(i) ) ./ ...
   (dx1(i).*dy2-dx2.*dy1(i)) ;
end

if size(line,2)==5 && size(edge,2)==5
    timeLine=line(:,5);
    timeEdge=edge(:,5);
    dt=timeEdge - timeLine;
    point = [xi' yi' dt];
    out = find(~onEdge(point, edge));
    point(out, :) = repmat([NaN NaN NaN], [length(out) 1]);
else
    point = [xi' yi'];
    out = find(~onEdge(point, edge));
    point(out, :) = repmat([NaN NaN], [length(out) 1]);
end

C.3.9 lineIntersect_miki.m

function [point]=lineIntersect_miki(edge1,edge2)
% calculates intersection and checks for parallel lines.
% also checks that the intersection point is actually on
% the line segment p1-p2

% calculate differences
xD1=edge1(:,3)-edge1(:,1);  
xD2=edge2(:,3)-edge2(:,1);  
yD1=edge1(:,4)-edge1(:,2);  
yD2=edge2(:,4)-edge2(:,2);  
xD3=edge1(:,1)-edge2(:,1);  
yD3=edge1(:,2)-edge2(:,2);

% calculate the lengths of the two lines
len1=sqrt(xD1.^2+yD1.^2);
len2=sqrt(xD2.^2+yD2.^2);

% find intersection Pt between two lines
div=yD2.*xD1-xD2.*yD1;
ua=(xD2.*yD3-yD2.*xD3)./div;
ub=(xD1.*yD3-yD1.*xD3)./div;
xint=edge1(:,1)+ua.*xD1;
yint=edge1(:,2)+ua.*yD1;

% calculate angle between the two lines.
dot=(xD1.*xD2+yD1.*yD2);
% dot product
deg=dot./(len1.*len2);

% if abs(angle)==1 then the lines are parallel,
% so no intersection is possible
xint(abs(deg)==1)=NaN;
yint(abs(deg)==1)=NaN;
% calculate the combined length of the two segments
% between Pt-p1 and Pt-p2
xD1=xint-edge1(:,1);
xD2=xint-edge1(:,3);
yD1=yint-edge1(:,2);
yD2=yint-edge1(:,4);
segmentLen1=sqrt(xD1.*xD1+yD1.*yD1)+sqrt(xD2.*xD2+yD2.*yD2);

% calculate the combined length of the two segments
% between Pt-p3 and Pt-p4
xD1=xint-edge2(:,1);
xD2=xint-edge2(:,3);
yD1=yint-edge2(:,2);
yD2=yint-edge2(:,4);
segmentLen2=sqrt(xD1.*xD1+yD1.*yD1)+sqrt(xD2.*xD2+yD2.*yD2);

% if the lengths of both sets of segments are the same as
% the lengths of the two lines the point is actually
% on the line segment.

% return the valid intersection (if the point isn’t on the line,
% return
% NaN)
point=[xint yint];
point((abs(len1-segmentLen1)>1e-4 | abs(len2-segmentLen2)>1e-4),:)=NaN;
% return NaN if the intersection is the origin of the second edge
point((abs(xD1)<1e-4 & abs(yD1)<1e-4),:)=NaN;

C.3.10  mask_dimensions.m
function [x_dim,y_dim]=mask_dimensions(mask)
% alçada de cada x
OX_mask=find(sum(mask,1)>0);
h_X=sum(mask,1);
x_dim=OX_mask-OX_mask(1);
y_dim=h_X(OX_mask);

% figure; plot(x_dim,y_dim)
% % amplada de cada y
% OY_mask=find(sum(mask,2)>0);
w_Y=sum(mask,2);
% subplot(2,1,2); plot(OY_mask-OY_mask(1),w_Y(OY_mask))

C.3.11  perimeter.m
function P=perimeter(varargin)
%PERIMETER
% P=perimeter(varargin) calculates the perimeter of a (x,y) series of
% points.
% Series of (x,y) points can be entered as a single argument or as two
% different arguments.
% There must be 3 or more (x,y) points.
% by Miquel Cubells i Gonzalvo 15/10/2009
if nargin==2
    x=varargin{1};
y=varargin{2};
elseif size(varargin{1},1)==2
    x=varargin{1}(1,:);'
y=varargin{1}(2,:)';
else
    x=varargin{1}(:,1);
y=varargin{1}(:,2);
end

xx=(x(2:end)-x(1:end-1)).^2;
yy=(y(2:end)-y(1:end-1)).^2;
P=sum(sqrt(xx+yy));

C.4 Altres

C.4.1 afegeix_imatges.m
% FUNCIÓ PER A AFEGIR IMATGES A UNA SEQÜÈNCIA D'IMATGES HOMOGRAFIADES
function [mat_T_nou,Timesec_nou,hora_nou,T_nou,z_im]=afegeix_imatges(mat_T,Timesec,hora,T,varargin)
% afegeix_imatges(mat_T,Timesec,hora,T): afegeix les imatges
% per homografiar carregant-les del directori de treball.
% afegeix_imatges(mat_T,Timesec,hora,T,mat_T_af,Timesec_af,hora_af.ini,T_af):
% afegeix les imatges sense homografiar i homografiades que se li entrin.

if nargin==4
    fname=dir('*.mat');
    for im=1:size(fname,1)
        load(fname(im).name,'*_DateTime*');
        varname=(strcat(fname(im).name(1:size(fname(im).name,2)-4),'
'_DateTime'));
        DateTime=eval(varname);
        Timesec_af(im,1)=DateTime(1,4)*3600+DateTime(1,5)*60+DateTime(1,6)+DatETIME(1,7)/1000;
        hora_af.ini{im,1}=strcat(num2str(DateTime(1,4),'%02d'),':',num2str(DateTime(1,5),'%02d'),'\',num2str(DateTime(1,6),'%02d'),'\',num2str(DateTime(1,7),'%03d'));
        clear(varname);
    end
else
    mat_T_af=varargin{1};
    Timesec_af=varargin{2};
    hora_af.ini=varargin{3};
    T_af=varargin{4};
end

[Timesec_af_s,order]=sort(Timesec_af);
z_im=[];
im=0;
while im<length(order)
im=im+1;
    if sum(find(Timesec==Timesec_af(order(im))))==0 % comprova que aquesta imatge realment no hi sigui!
        if Timesec_af_s(im)>max(Timesec)
            z_im=[z_im;length(Timesec)+order(im:end)]; % la resta d'imatges van al final
        end
    else
        if Timesec<Timesec_af_s(im)+order(im);
            z_im=[z_im;sum(Timesec<Timesec_af_s(im))+order(im)]; % troba la z de la imatge
        end
        else
            order(im:end-1)=order(im+1:end);
            order=order(1:end-1);
            im=im-1;
        end
    end
end

nova_dim=size(Timesec,1)+length(z_im);

Timesec_nou=zeros(nova_dim,1);
mat_T_nou=zeros(size(mat_T,1),size(mat_T,2),nova_dim);
T_nou=zeros(size(T,1),size(T,2),nova_dim);

for im=1:length(z_im)
    if nargin==4
        varname=fname(order(im)).name(1:size(fname(order(im)).name,2)-4);
        load(fname(order(im)).name,varname);
        clear(varname);
        else
            mat_T_nou(:,:,z_im(im))=flipud(eval(varname));
            T_nou(:,:,z_im(im))=T_af(:,:,order(im));
        end
    Timesec_nou(z_im(im))=Timesec_af(order(im));
    hora_nou.ordenada(z_im(im),1)=hora_af.ini{order(im),1};
end

if length(z_im)==1
    indexes_mat_T_nou=1:z_im-1;
else
    intervals_z_im=[1;z_im(1:end-1)+1] z_im-1;
    indexes_mat_T_nou=[];
    for z_int=1:length(z_im)
        indexes_mat_T_nou=[indexes_mat_T_nou
        intervals_z_im(z_int,1):intervals_z_im(z_int,2)];
    end
end
if isempty(z_im)==0
    if z_im(end)<nova_dim
        indexes_mat_T_nou=[indexes_mat_T_nou (z_im(end)+1):nova_dim];
    end
    Timesec_nou(indexes_mat_T_nou)=Timesec(1:end);
    hora_nou.ordenada(indexes_mat_T_nou,1)=hora.ordenada(1:end,1);
```matlab
mat_T_nou(:,:,indexs_mat_T_nou)=mat_T(:,:,1:end);
T_nou(:,:,indexs_mat_T_nou)=T(:,:,1:end);
else
    Timesec_nou=Timesec;
    hora_nou.ordenada=hora.ordenada;
    mat_T_nou=mat_T;
    T_nou=T;
end

C.4.2 annotateplot.m

function annotateplot(dim,V,strcell,varargin)
axisvec=axis;
if size(V,1)<size(V,2)
    V=V';
end
% VV=reshape(repmat(V',2,1),[2*size(V,1) 1]);
if dim==1
    % plot(VV,repmat([axisvec(3);axisvec(4)],size(V)),'k', 'LineStyle', '--')
    upspace=0.05*abs(axisvec(4)-axisvec(3));
    for item=1:length(V);
        hold on;
        plot([V(item),V(item)],[axisvec(3),axisvec(4)],'Color','k','Line','--')
        text(V(item),axisvec(4)+upspace,strcell{item},'HorizontalAlignment','center','FontSize',8)
    end
elseif dim==2
    for item=1:length(V);
        hold on;
        plot([axisvec(1),axisvec(2)],[V(item),V(item)],'Color','k','Line','--')
    end
end
function [h,varargout]=compassPlot(R, X0, Y0, north_phase,directions,varargin)
% CompassPlot
%   compassPlot(R, X0, Y0, north_phase,directions,varargin)
%   plots a compass indicating north and any desired directions.
%   
%   R: radius of the circumference
%   X0: x-coordinate of circumference centre
%   Y0: y-coordinate of circumference centre
%   north_phase: phase of North expressed in clockwise degrees (0° faces North upwards)
%   directions: directions related to North to draw
%   varargin{1}: strings containing the color of the rows
%   varargin{2}: cell of strings containing the legend of each arrow
%   
%   by Miquel Cubells i Gonzalvo 08/10/2009
```
% Define and draw the circumference
N=100;
[xcirc,ycirc,h]=createCircumference(R, X0, Y0,N,1);
varargout{1}=xcirc;
varargout{2}=ycirc;

if nargin>3
    % Plot North arrow
    xnorth=[X0 cosd(90-north_phase)*1.75*R+X0];
    ynorth=[Y0 sind(90-north_phase)*1.75*R+Y0];
    hold on;
    drawArrow_miki(xnorth(1),ynorth(1),xnorth(2),ynorth(2),R/5,R/5,1);
    text(cosd(90-north_phase)*2.1*R+X0,sind(90-north_phase)*2.1*R+Y0,'N',...'
HorizontalAlignment','center');
    axis off
    % Plot desired arrows
    if nargin>4
        if size(directions,2)>size(directions,1)
            directions=directions';
        end
        if isempty(varargin)==0
            color=varargin{1};
        else
            color='r';
        end
        if size(directions,2)==1
            xarrow=[X0.*ones(size(directions)) cosd(90-north_phase-direc...}
            yarrow=[Y0.*ones(size(directions)) sind(90-north_phase-direc...}
            drawArrow_miki(xarrow(:,1),yarrow(:,1),xarrow(:,2),yarrow(:,...
',R/5,R/5,1,color);
        elseif size(directions,2)>1
            error('MATLAB:compassPlot:InvalidInput', 'directions variable has to be a 1-dimension array.');
        end
        if nargin>5
            vec_str=varargin{2};
            text(cosd(90-north_phase-direc...}
            text(cosd(90-north_phase-direc...}
            end
    end
end
hold off

C.4.4 drawArrow_miki.m
function varargout = drawArrow_miki(varargin)
%DRAWARROW draw an arrow on the current axis
% usage:
% DRAWARROW(x1, y1, x2, y2) draw an arrow between the points (x1 y1) and
% (x2 y2).
DRAWARROW([x1 y1 x2 y2]) also works.

DRAWARROW(..., L, W)
% specify length and width of the arrow.

DRAWARROW(..., L, W, TYPE)
% also specify arrow type. TYPE can be one of the following :
% 0 : draw only two strokes
% 1 : fill a triangle
% .5 : draw a half arrow (try it to see ...)

Arguments can be single values or array of size [N*1]. In this case,
the function draws multiple arrows.

H = DRAWARROW(...) return handle(s) to created edges(s)

---------
author : David Legland
INRA - TPV URPOI - BIA IMASTE
created the 11/11/2004 from drawEdge

Adapted by: Miquel Cubells i Gonzalvo

if isempty(varargin)
    error('should specify at least one argument');
end

if ischar(varargin{length(varargin)})==0
    color='k';
else
    color=varargin{length(varargin)};
end

% parse arrow coordinate
var = varargin{1};
if size(var, 2)==4
    x1 = var(:,1);
    y1 = var(:,2);
    x2 = var(:,3);
    y2 = var(:,4);
    varargin = varargin(2:end);
elseif length(varargin)>3
    x1 = varargin{1};
y1 = varargin{2};
x2 = varargin{3};
y2 = varargin{4};
varargin = varargin(2:end);
else
    error('wrong number of arguments, please read the doc');
end

l = 10*size(size(x1));
Desenvolupament d'una metodologia basada en termografia infraroja...

\[ w = 5 \times \text{ones} \left( \text{size}(x1) \right); \]
\[ h = \text{zeros} \left( \text{size}(x1) \right); \]

% extract length of arrow
if ~isempty(varargin)
    l = varargin{1};
    if length(x1) > length(l)
        l = l(1) \times \text{ones} \left( \text{size}(x1) \right);
    end
end

% extract width of arrow
if length(varargin) > 1
    w = varargin{2};
    if length(x1) > length(w)
        w = w(1) \times \text{ones} \left( \text{size}(x1) \right);
    end
end

% extract 'ratio' of arrow
if length(varargin) > 2
    h = varargin{3};
    if length(x1) > length(h)
        h = h(1) \times \text{ones} \left( \text{size}(x1) \right);
    end
end

hold on;
axis equal;

% angle of the edge
theta = \text{atan2} \left( y2 - y1, x2 - x1 \right);

% point on the 'left'
xa1 = x2 - l \times \cos(theta) - w \times \sin(theta)/2;
ya1 = y2 - l \times \sin(theta) + w \times \cos(theta)/2;
% point on the 'right'
xa2 = x2 - l \times \cos(theta) + w \times \sin(theta)/2;
ya2 = y2 - l \times \sin(theta) - w \times \cos(theta)/2;
% point on the middle of the arrow
xa3 = x2 - l \times \cos(theta) \times h;
ya3 = y2 - l \times \sin(theta) \times h;

% draw main edge
hl = \text{line}([x1'; x2'], [y1'; y2'], 'color', color);
for k=1:length(x1)
    set(get(get(hl(k),'Annotation'),'LegendInformation'),'IconDisplayStyle ','off'); % Exclude line from legend
end
% draw only 2 wings
ind = find(h==0);
line([xa1(ind)'; x2(ind)'], [ya1(ind)'; y2(ind)'], 'color', color);
line([xa2(ind)'; x2(ind)'], [ya2(ind)'; y2(ind)'], 'color', color);
% draw a full arrow
ind = find(h>0);
hl = \text{patch}([x2(ind) xa1(ind) xa3(ind) xa2(ind) x2(ind)]', ...
                 [y2(ind) ya1(ind) ya3(ind) ya2(ind) y2(ind)]', color);
C.4.5 elimina_imatge.m
%% FUNCIÓ PER A ELIMINAR UNA IMATGE HOMOGRafiADA CORRECTAMENT
function [mat_T_nou,Timesec_nou]=elimina_imatge(mat_T,T,Timesec,z)
if z<size(mat_T,3)
    mat_T_nou=cat(3,mat_T(:,:,1:z-1),mat_T(:,:,z+1:size(mat_T,3)));
    T_nou=T(:,:,1:z-1);
    Timesec_nou=cat(1,Timesec(1:z-1,1),Timesec(z+1:size(Timesec,1),1));
elseif z==size(mat_T,3)
    mat_T_nou=mat_T(:,:,1:z-1);
    T_nou=T(:,:,1:z-1);
    Timesec_nou=Timesec(1:z-1,1);
end

C.4.6 getnicefigurelocation.m
function figure_size = getnicefigurelocation(figure_size, figure_units)
% adjust the specified figure position to fig nicely over GCBF
% or into the upper 3rd of the screen
% Adapted by Miquel Cubells from:
% Copyright 1999-2006 The MathWorks, Inc.
% $Revision: 1.1.6.3 $

%%%%%% PLEASE NOTE %%%%%%%%%
%%%%%% This file has also been copied into:
%%%%%% matlab/toolbox/ident/idguis
%%%%%% If this functionality is changed, please
%%%%%% change it also in idguis.
%%%%%% PLEASE NOTE %%%%%%%%%

parentHandle = gcbf;
propName = 'Position';
if isempty(parentHandle)
    parentHandle = 0;
    propName = 'ScreenSize';
end

old_u = get(parentHandle,'Units');
set(parentHandle,'Units',figure_units);
container_size=get(parentHandle,propName);
set(parentHandle,'Units',old_u);

figure_size(1) = container_size(1) + 1/2*(container_size(3) - figure_size(3));
figure_size(2) = container_size(2) + 1/2*(container_size(4) - figure_size(4));
Desenvolupament d'una metodologia basada en termografia infraroja...

C.4.7 resizefigures.m
function resizefigures(varargin)
if isempty(varargin)
    sizefactor=1;
elseif nargin==1
    sizefactor=varargin(1);
end
fullscreen=get(0,'Screensize');
set(gcf,'Position',sizefactor.*[3 34 fullscreen(3)-111]);
if sizefactor<0.9
    set(gcf,'Position',getnicefigurelocation(get(gcf,'Position'),get(gcf,'Units')));
end

C.4.8 afegir_tdi_tdf.m
%% Script per afegir TDI i TDF a una seqüència
T_prov=T;
Timesec_prov=Timesec;
hora_prov=hora;
mat_T_prov=mat_T;

T_tditdf_filename='homografies_z2.mat';
load(T_tditdf_filename,'mat_T','T','Timesec','hora');

mat_T_af=mat_T(:,:,1:2);
T_af=T(:,:,1:2);
Timesec_af=Timesec(1:2);
hora_af.ini=hora.ordenada(1:2);

[mat_T,Timesec,hora,T,z_im]=afegeix_imatges(mat_T_prov,Timesec_prov,hora_prov,T_prov,mat_T_af,Timesec_af,hora_af.ini,T_af);

C.4.9 mostreig_seqmat.m
%% SCRIPT PER A EXTRoure IMATGES D'Uuna SEQüÈNCIA CONVERTIDA A .MAT
%========================================================================
% Carregar les matrius i ordenar-les cronològicament
%========================================================================
clear all
ir_basename='as2w_'; % nom base dels fitxers .mat
ir_period=20; % període en segons de captació d'imatge (es desarà una
imatge per període i la resta s'esborraràn permanentment del directori
initial_frame=1; % primer frame que ens interessa

ir_basename_ext=strcat(ir_basename,'*.mat');
ir_fname=dir(ir_basename_ext);
numframes=size(ir.fname,1);
for numframe=1:numframes
    if numframe<initial_frame
        numframestr=num2str(numframe,'%4d');
        current_fname=(strcat(ir.basename,numframestr,'.mat'));
        delete(current_fname)
    elseif numframe==initial_frame
        last_valid_frame=initial_frame;
    elseif numframe==last_valid_frame+round(ir_period*cam_freq)
        last_valid_frame=numframe;
    else
        numframestr=num2str(numframe);
        current_fname=(strcat(ir.basename,numframestr,'.mat'));
        delete(current_fname)
    end
end

C.4.10 move_image.m

function imatge_nova=move_image(imatge_original,files,columnes)
imatge_nova=imatge_original;

% pujar(+) i baixar(-) la imatge (axis xy)
if files<0
    imatge_nova(1:size(imatge_original,1)+files,:)=imatge_original(-files+1:size(imatge_original,1),:);
elseif files>0
    imatge_nova(files+1:size(imatge_original,1),:)=imatge_original(1:size(imatge_original,1)-files,:);
end

% desplaçar dreta(+) i esquerra(-) la imatge (axis xy)
if columnes<0
    imatge_nova(:,1:size(imatge_original,2)+columnes)=imatge_nova(:,columnes+1:size(imatge_original,2));
elseif columnes>0
    imatge_nova(:,columnes+1:size(imatge_original,2))=imatge_nova(:,1:size(imatge_original,2)-columnes);
end

C.4.11 rateofspread.m

function [ros, varargout]=rateofspread(isochrons)

% Split each isochron edge between pairs of points by means of linear
% splines interpolation and find its normal line.
N=4; % Number of divisions
ros=[];
pos_z_ros=nan(size(isochrons.timesec,1)-1,2);
for i=1:(size(isochrons.timesec,1)-1)
    pos_z_ros(i,1:2)=[i length(ros)+1];
    xy=vertcat(isochrons.xy{i,:});
    [r,c,m]=interpoolEdge(xy(:,1),xy(:,2),N);
    line=createLine_miki(r,c,-1./m); % create normal lines
line(:,end+1)=repmat(isochrons.timesec{i},size(line,1),1);

% Create an edge matrix containing [x1 y1 x2 y2 timesec] in each row
edge=[];

for j=1:isochrons.numsections{i+1}
    edge=[edge; isochrons.xy{i+1,j}(1:end-1,:)
        isochrons.xy{i+1,j}(2:end,:)];
end

numedges=size(edge,1);
edge(:,end+1)=isochrons.timesec{i+1}.*ones(numedges,1);

if numedges>1
    % Prepare line and edge in form of matrices
    numlines=size(line,1);
    numedges=size(edge,1);

    line_mat=reshape(repmat(line',numedges,1),size(line,2),size(line,1)*numedges)';
    edge_mat=repmat(edge,numlines,1);

    % Find intersection between lines and edges and calculate distance
    point=intersectLineEdge_miki(line_mat, edge_mat);
    point(:,4)=hypot(point(:,1)-line_mat(:,1),point(:,2)-
        line_mat(:,2));

    % Select the correct solution (minimum distance)
    point=shiftdata(reshape(point',4,numedges,numlines),2);
    [mindist,minrow]=min(point(:,4,:));
    point=point(reshape(0:numedges:numel(point)-
        1,numlines)'+repmat(minrow(:),1,4));

    % Clear points belonging to the opposite flank by clearing vectors
    % crossing its own origin isochron or any previous one.
    edge1=[];
    if i>1
        for ii=1:i
            for j=1:isochrons.numsections{ii}
                edge1=[edge1; isochrons.xy{ii,j}(1:end-1,:) 
                    isochrons.xy{ii,j}(2:end,:)];
            end
        end
    elseif i==1
        for j=1:isochrons.numsections{i}
            edge1=[edge1; isochrons.xy{i,j}(1:end-1,:) 
                isochrons.xy{i,j}(2:end,:)];
        end
    end
    edge2=[line(:,1:2) point(:,1:2)];
    numedges1=size(edge1,1);
    numedges2=size(edge2,1);
    edgel_mat=repmat(edge1,numedges2,1);
edge2_mat=reshape(repmat(edge2',numedges1,1),size(edge2,2),size(edge2,1)*numedges1);  
    selfintersection=lineIntersect_miki(edge1_mat,edge2_mat);  
    find(isnan(selfintersection(:,1))==0);  

selfintersection_ind=find(isnan(selfintersection(:,1))+isinf(selfintersection(:,1))==0);  
    point(ceil(selfintersection_ind./numedges1),:)=NaN;  
    ros=[ros; line(:,1:2) point];  
end  
end  
atg_ros=atan2((ros(:,4)-ros(:,2)),(ros(:,3)-(ros(:,1))));  
    ros(:,7:10)=[ros(:,6)./ros(:,5) cos(atg_ros) sin(atg_ros) atg_ros];  
varargout{1}=pos_z_ros;  

C.4.12 zones5.m  
function [f,g,r,b,ub]=zones5(T_matrix,V_criterion)  
    "  
    % INPUT  
    % T_matrix: Temperature matrix to be zoned  
    % V_criterion: (Optional)  
    % Segmentation vector containg the minimum temperature  
    % criterion to zone T_matrix: [T_ub;T_h;T_r;T_g;T_f]  
    % Default value is V_criterion=[303;360;425;600;700]  
    % OUTPUT  
    % f,g,r,b,ub: absolut indexes of T_matrix cells correspoding to  
    % flaming, glowing,residual, burned or preheating or dropped,  
    % unburned, respectively.  
    if nargin==1  
        V_criterion=[303;360;425;600;700]; %vector de segmentació:  
            [flamming, glowing, residual, burned or preheating or dropped,  
            unburned];  
    end  
    f=find(T_matrix>=V_criterion(5));  
    % Flaming  
    g=find((T_matrix>=V_criterion(4))&(T_matrix)<V_criterion(5));  
    % Glowing  
    r=find((T_matrix>=V_criterion(3))&(T_matrix)<V_criterion(4));  
    % Residual  
    b=find((T_matrix>=V_criterion(2))&(T_matrix)<V_criterion(3));  
    % Burned  
    ub=find(T_matrix>=V_criterion(1)&T_matrix<V_criterion(2));  
    % Unburned
C.4.13 zones_data5.m

function [zones_distro,zones_percentage]=zones_data5(f,g,r,b,ub)

% INPUT
% f,g,r,b,ub: absolut indexes of T_matrix cells correspoding to
% flamming, glowing,residual, burned or preheating or dropped, unburned, respectively.

% OUTPUT
% zones_distro: Area [pixels] of flamming, glowing,residual, burned or preheating or dropped, unburned zones and the whole analysed zone, respectively
% zones_percentage: Percentage of the analysed zone covering flamming, glowing, residual, burned or preheating or dropped, unburned

zones_percentage=zeros(1,5);
zones_distro=zeros(1,6);

% Area [pixels] flamming, glowing, residual, burned, unburned
zones_distro(1,1)=length(f);
zones_distro(1,2)=length(g);
zones_distro(1,3)=length(r);
zones_distro(1,4)=length(b);
zones_distro(1,5)=length(ub);

% Area [pixels] of drop zone
zones_distro(1,6)=sum(zones_distro(1,1:5));

% Percentage of the drop zone flamming, glowing, residual, burned, unburned
zones_percentage(1,1)=100*(zones_distro(1,1)/zones_distro(1,6));
zones_percentage(1,2)=100*(zones_distro(1,2)/zones_distro(1,6));
zones_percentage(1,3)=100*(zones_distro(1,3)/zones_distro(1,6));
zones_percentage(1,4)=100*(zones_distro(1,4)/zones_distro(1,6));
zones_percentage(1,5)=100*(zones_distro(1,5)/zones_distro(1,6));

C.4.14 zoom_mask.m

function [zoom_rows,zoom_columns]=zoom_mask(mask,margin_rows,margin_cols)
if nargin==2
    margin_cols=margin_rows;
end
rows_drop=find(sum(mask,2)>0);
columns_drop=find(sum(mask,1)>0);

% Zoom area
zoom_rows=rows_drop(1)-margin_rows:rows_drop(size(rows_drop,1))+margin_rows;
zoom_columns=1:columns_drop(size(columns_drop,1))+margin_rows;
if zoom_rows(1)<0;
    zoom_rows=1:rows_drop(size(rows_drop,1))+margin_rows;
end
if zoom_rows(size(zoom_rows,2))>size(mask,1)
zoom_rows = rows_drop(1) - margin_rows: size(mask, 1);
end

zoom_columns = columns_drop(1) - margin_cols: columns_drop(size(columns_drop, 2)) + margin_cols;
if zoom_columns(1) < 0;
    zoom_columns = 1: columns_drop(size(columns_drop, 2)) + margin_cols;
end

if zoom_columns(size(zoom_columns, 2)) > size(mask, 2)
    zoom_columns = columns_drop(1) - margin_cols: size(mask, 2);
end
C.5 Funcions d'autoria aliena
El nom dels creadors figura als codis.

C.5.1 plotirconvtemp2.m

%=====================================================================
%   Plot imatge IR i modifica escala colors
%   Paràmetres d’entrada:
%   matrix: arxiu '.mat'
%   Tim: valor llindar de temperatura per a poder visualitzar
%   referències en
%   la imatge
%   Exemple:
%   plotirconvtemp(mat3464,350);
%   Versió adaptada de la funció 'plotdataimgB.m' de Miguel Muñoz
%   Versió 2 de la funció plotirconvtemp.m
%   YPR @ 2008
%=====================================================================

function plotirconvtemp2(matrix,Tim)
  mnewtemp=matrix;
  mnewtemp(find(mnewtemp>=Tim))=Tim;
  mx=max(max(mnewtemp));
  mi=min(min(mnewtemp));
  df=mx-mi;
  if ~(df==0)
    mnewtemp=(mnewtemp-mi)./df;
  end
  map=ironcolormap;
  cx=[0 1];
  matrix=full(mnewtemp);
  hFig = figure('Toolbar','none', ...
    'Menubar','none');
  hIm=imshow(mnewtemp);
  hSP = imscrollpanel(hFig,hIm);
  set(hSP,'Units','normalized');
  api = iptgetapi(hSP);
  api.setMagnification(4);
  set(gcf,'Units','normalized','Position',[0.005 0.05 0.99 0.915])
  ha=gca;
  caxis(cx);
  colormap(map);

C.5.2 checkargs.m

% Function to check argument values and set defaults
function [x1, x2] = checkargs(arg);
if length(arg) == 2
    x1 = arg{1};
    x2 = arg{2};
    if ~all(size(x1)==size(x2))
        error('x1 and x2 must have the same size');
    elseif size(x1,1) ~= 3
        error('x1 and x2 must be 3xN');
    end

elseif length(arg) == 1
    if size(arg{1},1) ~= 6
        error('Single argument x must be 6xN');
    else
        x1 = arg{1}(1:3,:);
        x2 = arg{1}(4:6,:);
    end
else
    error('Wrong number of arguments supplied');
end

C.5.3 intline.m
function [x,y] = intline(x1, x2, y1, y2)

%INTLINE Integer-coordinate line drawing algorithm.
%   [X, Y] = INTLINE(X1, X2, Y1, Y2) computes an
% approximation to the line segment joining (X1, Y1) and
% (X2, Y2) with integer coordinates.  X1, X2, Y1, and Y2
% should be integers.  INTLINE is reversible; that is,
% INTLINE(X1, X2, Y1, Y2) produces the same results as
% FLIPUD(INTLINE(X2, X1, Y2, Y1)).

% Copyright 1993-2003 The MathWorks, Inc.
% $Revision: 5.11.4.2 $  $Date: 2004/08/10 01:45:22 $

dx = abs(x2 - x1);
dy = abs(y2 - y1);

% Check for degenerate case.
if ((dx == 0) && (dy == 0))
    x = x1;
    y = y1;
    return;
end

flip = 0;
if (dx >= dy)
    if (x1 > x2)
% Always "draw" from left to right.
        t = x1; x1 = x2; x2 = t;
        t = y1; y1 = y2; y2 = t;
        flip = 1;
    end
    m = (y2 - y1)/(x2 - x1);
    x = (x1:x2).';
    y = round(y1 + m*(x - x1));
else
    if (y1 > y2)
% Always "draw" from bottom to top.
        t = y1; y1 = y2; y2 = t;
        t = x1; x1 = x2; x2 = t;
        m = (x2 - x1)/(y2 - y1);
        x = (y1:y2).';
        y = round(x + m*(y - y1));
    end
end

if flip
    [x y] = flipud([x y]);
else
    [x y] = [x y];
end
end
\begin{verbatim}
t = x1; x1 = x2; x2 = t;
t = y1; y1 = y2; y2 = t;
flip = 1;
end
m = (x2 - x1)/(y2 - y1);
y = (y1:y2).';
x = round(x1 + m*(y - y1));
end

if (flip)
x = flipud(x);
y = flipud(y);
end
\end{verbatim}

C.5.4  
\texttt{normalise2dpts.m}

\texttt{function \[newpts, T\] = normalise2dpts(pts)}

\texttt{\% \texttt{NORMALISE2DPTS} - normalises 2D homogeneous points}
\texttt{\%}  
\texttt{\% Function translates and normalises a set of 2D homogeneous points}
\texttt{\% so that their centroid is at the origin and their mean distance from}
\texttt{\% the origin is sqrt(2). This process typically improves the}
\texttt{\% conditioning of any equations used to solve homographies,}
\texttt{\% fundamental}
\texttt{\% matrices etc.}
\texttt{\%}  
\texttt{\% Usage: \[newpts, T\] = normalise2dpts(pts)}
\texttt{\%}  
\texttt{\% Argument:}
\texttt{\% \ \ pts - 3xN array of 2D homogeneous coordinates}
\texttt{\%}  
\texttt{\% Returns:}
\texttt{\% \ \ newpts - 3xN array of transformed 2D homogeneous coordinates.}
\texttt{\% The}
\texttt{\% \ \ scaling parameter is normalised to 1 unless the point is}
\texttt{\% \ \ at}
\texttt{\% \ \ infinity.}
\texttt{\% \ \ T \ \ - The 3x3 transformation matrix, newpts = T*pts}
\texttt{\%}  
\texttt{\% If there are some points at infinity the normalisation transform}
\texttt{\% is calculated using just the finite points. Being a scaling and}
\texttt{\% translating transform this will not affect the points at infinity.}
\texttt{\%}  
\texttt{Peter Kovesi}
\texttt{\% School of Computer Science & Software Engineering}
\texttt{\% The University of Western Australia}
\texttt{\% pk at csse uwa edu au}
\texttt{\% http://www.csse.uwa.edu.au/~pk}
\texttt{\%}  
\texttt{\% May 2003 \ \ - Original version}
\texttt{\% February 2004 \ \ - Modified to deal with points at infinity.}
\texttt{\% December 2008 \ \ - meandist calculation modified to work with Octave}
\texttt{\% 3.0.1}
\texttt{\% \ \ (thanks to Ron Parr)
error('pts must be 3xN');
end

% Find the indices of the points that are not at infinity
finiteind = find(abs(pts(3,:)) > eps);

if length(finiteind) ~= size(pts,2)
    warning('Some points are at infinity');
end

% For the finite points ensure homogeneous coords have scale of 1
pts(1,finiteind) = pts(1,finiteind)./pts(3,finiteind);
pts(2,finiteind) = pts(2,finiteind)./pts(3,finiteind);
pts(3,finiteind) = 1;

        % Centroid of finite points
        c = mean(pts(1:2,finiteind)');
        newp(1,finiteind) = pts(1,finiteind)-c(1); % Shift origin to centroid.
        newp(2,finiteind) = pts(2,finiteind)-c(2);

dist = sqrt(newp(1,finiteind).^2 + newp(2,finiteind).^2);
meandist = mean(dist(:));  % Ensure dist is a column vector for Octave 3.0.1
scale = sqrt(2)/meandist;
T = [scale 0 -scale*c(1)
     0 scale -scale*c(2)
     0 0 1 ];

newpts = T*pts;

C.5.5  getedges.m
function [r,c]=getedges(x,y,closed)
if nargin==2
    closed=1;
end
x=floor(x-0.5)+1;
y=floor(y-0.5)+1;

if closed==1
    if ~(x(1)==x(end)) || ~(y(1)==y(end))
        y(end+1)=y(1);
        x(end+1)=x(1);
    end
end

num_segments = length(x) - 1;
x_segments = cell(num_segments,1);
y_segments = cell(num_segments,1);
for k = 1:num_segments
    [x_segments{k},y_segments{k}] = intline(x(k),x(k+1),y(k),y(k+1));
end
% Concatenate segment vertices.
c = cat(1,x_segments{:});
r = cat(1,y_segments{:});

C.5.6  homography2d.m
% HOMOGRAPHY2D - computes 2D homography
% Usage:   H = homography2d(x1, x2)  
%          H = homography2d(x)  
% Arguments:  
%    x1 - 3xN set of homogeneous points  
%    x2 - 3xN set of homogeneous points such that x1<>x2  
%    x - If a single argument is supplied it is assumed that it  
%         is in the form x = [x1; x2]  
% Returns:  
%    H - the 3x3 homography such that x2 = H*x1  
% This code follows the normalised direct linear transformation  
% algorithm given by Hartley and Zisserman "Multiple View Geometry in  
% Computer Vision" p92.  
%  
% Peter Kovesi  
% School of Computer Science & Software Engineering  
% The University of Western Australia  
% pk at csse uwa edu au  
% http://www.csse.uwa.edu.au/~pk  
% May 2003 - Original version.  
% Feb 2004 - Single argument allowed for to enable use with RANSAC.  
% Feb 2005 - SVD changed to 'Economy' decomposition (thanks to Paul O'Leary)

function H = homography2d(varargin)
    [x1, x2] = checkargs(varargin{:});
    % Attempt to normalise each set of points so that the origin  
    % is at centroid and mean distance from origin is sqrt(2).  
    [x1, T1] = normalise2dpts(x1);
    [x2, T2] = normalise2dpts(x2);
    % Note that it may have not been possible to normalise  
    % the points if one was at infinity so the following does not  
    % assume that scale parameter w = 1.
    Npts = length(x1);
    A = zeros(3*Npts,9);
    O = [0 0 0];  
    for n = 1:Npts
        X = x1(:,n)';
        x = x2(1,n); y = x2(2,n); w = x2(3,n);
        A(3*n-2,:) = [ O  -w*X  y*X];
        A(3*n-1,:) = [ w*X  O  -x*X];
    end  

% Concatenate segment vertices.
c = cat(1,x_segments{:});
r = cat(1,y_segments{:});
\[ A(3*n ,:) = \begin{bmatrix} -y & x & 0 \end{bmatrix}; \]

end

\[ [U,D,V] = \text{svd}(A,0); \] 'Economy' decomposition for speed

% Extract homography
\[ H = \text{reshape}(V(:,9),3,3)'; \]

% Denormalise
\[ H = T2\backslash H\cdot T1; \]

C.5.7 onEdge.m

function b = onEdge(point, edge)
% ONEDGE test if a point belongs to an edge
% usage :
% b = onEdge(POINT, EDGE)
% with POINT being [xp yp], and EDGE being [x1 y1 x2 y2].
% ------
% author : David Legland
% INRA - TPV URPOI - BIA IMASTE
% created the 31/10/2003.
% HISTORY
% 11/03/2004 : change input format : edge is [x1 y1 x2 y2].
% 17/01/2005 : if test N edges with N points, return N boolean.
% 21/01/2005 : normalize test for colinearity, so enhance precision

Np = size(point, 1);
Ne = size(edge, 1);

if Np==1 || Ne==1
\[ x0 = \text{repmat}(\text{edge}(:,1)', Np, 1); \]
\[ y0 = \text{repmat}(\text{edge}(:,2)', Np, 1); \]
\[ dx = \text{repmat}(\text{edge}(:,3)', Np, 1)-x0; \]
\[ dy = \text{repmat}(\text{edge}(:,4)', Np, 1)-y0; \]
\[ xp = \text{repmat}(\text{point}(:,1), 1, Ne); \]
\[ yp = \text{repmat}(\text{point}(:,2), 1, Ne); \]
elseif Np==Ne
\[ x0 = \text{edge}(:,1); \]
\[ y0 = \text{edge}(:,2); \]
\[ dx = \text{edge}(:,3)-x0; \]
\[ dy = \text{edge}(:,4)-y0; \]
\[ xp = \text{point}(:,1); \]
\[ yp = \text{point}(:,2); \]
end

% test if lines are colinear
\[ b1 = \text{abs}((xp-x0).*dy - (yp-y0).*dx)./(dx.*dx+dy.*dy)<1e-13; \]

\[ \text{ind} = \text{abs}(dx)>\text{abs}(dy); \]
\[ t = \text{zeros}(	ext{max(Np, Ne), 1}); \]
\[ t\{\text{ind}\} = (xp\{\text{ind}\}-x0\{\text{ind}\})./dx(\text{ind}); \]
\[ t\{-\text{ind}\} = (yp\{-\text{ind}\}-y0\{-\text{ind}\})./dy(-\text{ind}); \]
\[ b = t>1e-14 & t<1e-14 & b1; \]
Desenvolupament d’una metodología basada en termografia infrarroja...
D  INFORMES DE LES DESCÀRREGUES DEL TERRENY AS2-E

A les següents pàgines s’adjunten els informes de les descàrregues efectuades per al terreny AS2-E tal com les realitza l’aplicació implementada.
INFORME D'AVALUACIÓ DE DESCÀRREGA AÈRIA

Aquest informe està orientat a facilitar la resposta de les qüestions clau proposades a Desenvolupament d'una metodologia basada en termografia infraroja per a l'anàlisi quantitativa de productes químics en l'extinció d'incendis forestals.

Contents

- Detalls de la descàrrega avaluada
- Qüestió 1
- Qüestió 2
- Qüestió 3
- Qüestió 4
- Qüestió 5
- Qüestió 6
- Qüestió 7

Informe generat el dia 27/10/2009 a les 17:29:19

Detalls de la descàrrega avaluada

Id. Experiment: AS2E
Id. Descàrrega: Drop 1

Qüestió 1

La descàrrega ha assolit a l'objectiu desitjat? (Foc puntual, cap, flanc, assistència dels mitjans terrestres).

Hi ha 1 foc secundari a l'instant tdi.
Qüestió 2

L'objectiu triat era el més adequat? (S'ha fixat com a objectiu la zona més activa del perímetre de l'incendi o del flanc triat?)

Temperatura aparent entre d1 i d2 respecte Tmin i Tmàx de tot el perímetre: Quartil 2 (Percentil 46)
Velocitat de propagació entre d1 i d2 respecte Vp.min i Vp.màx de tot el perímetre: Quartil 3 (Percentil 52)
Qüestió 3
La descàrrega ha assolit la millor orientació en relació amb el vent, els paràmetres de vol i la direcció de propagació del foc?

Àrea de la descàrrega: 3555 m²
Dosi efectiva: 0.90 l/m²
Perímetre de la descàrrega: 349 m
Centroide: (485601 Easting, 6042319 Northing)
Longitud màxima: 137 m
Amplada màxima: 36 m (Situada a 0.56·Lmax)
Amplada mitjana del tram 0.25·Lmax - 0.75·Lmax: 31.4 m (Desviació estàndard: 3.49 m)
Longitud L': 94 m
Direcció de Lmax: 142º respecte el nord
Direcció de vol: 157º respecte el nord
Direcció del vent: 111º respecte el nord
Direcció de propagació: 48º respecte el nord
Qüestió 4

La descàrrega ha assolit el tram pretès del perímetre? Hi ha zones ja cremades o combustible sense cremar?
Un 25.0% dels punts tals que 0.9<T<Tmàx ha caigut en un radi de 10 m del centroide
Zona de segmentació de temperatura més alta de la descàrrega: Flama
Distància entre el centroide de la descàrrega i el de la zona Flama: 14 m,
queda, doncs, fora del radi de 10 m del centroide de la descàrrega.
Qüestió 5
Quin percentatge de la longitud de la descàrrega ha resultat més efectiu?
Qüestió 6

S'ha solapat correctament la descàrrega en estudi amb la descàrrega precedent?
Aquesta descàrrega no té cap descàrrega precedent.

Qüestió 7

Hi ha zones sense cobrir dins la descàrrega?

Superfície sense cobrir: 9 m² (un 0.25 % de la superfície de la descàrrega)
INFORME D'AVALUACIÓ DE DESCÀRREGA AÈRIA

Aquest informe està orientat a facilitar la resposta de les qüestions clau proposades a Desenvolupament d'una metodologia basada en termografia infraroja per a l'anàlisi quantitativa de productes químics en l'extinció d'incendis forestals.

Contents

- Detalls de la descàrrega avaluada
- Qüestió 1
- Qüestió 2
- Qüestió 3
- Qüestió 4
- Qüestió 5
- Qüestió 6
- Qüestió 7

Informe generat el dia 27/10/2009 a les 17:22:32

Detalls de la descàrrega avaluada

Id. Experiment: AS2E
Id. Descàrrega: Drop 2

Qüestió 1

La descàrrega ha assolit a l'objectiu desitjat? (Foc puntual, cap, flanc, assistència dels mitjans terrestres).

Hi ha 1 foc secundari a l'instant tdi.
Qüestió 2

L'objectiu triat era el més adequat? (S'ha fixat com a objectiu la zona més activa del perímetre de l'incendi o del flanc triat?)

Temperatura aparent entre d1 i d2 respecte Tmin i Tmàx de tot el perímetre:
Quartil 2 (Percentil 48)

Velocitat de propagació entre d1 i d2 respecte Vp.min i Vp.màx de tot el perímetre:
Quartil 2 (Percentil 50)
Qüestió 3

La descàrrega ha assolit la millor orientació en relació amb el vent, els paràmetres de vol i la direcció de propagació del foc?

Àrea de la descàrrega: 4680 m²
Dosi efectiva: 0.68 l/m²
Perímetre de la descàrrega: 425 m
Centroide: (485721 Easting, 6042368 Northing)
Longitud màxima: 138 m
Amplada màxima: 60 m (Situada a 0.67·Lmax)
Amplada mitjana del tram 0.25·Lmax – 0.75·Lmax: 47.7 m (Desviació estàndard: 9.44 m)
Longitud L’: 83 m
Direcció de Lmax: 134º respecte el nord
Direcció de vol: 155º respecte el nord
Direcció del vent: 111º respecte el nord
Direcció de propagació: 41º respecte el nord
Qüestió 4

La descàrrega ha assolit el tram pretès del perímetre? Hi ha zones ja cremades o combustible sense cremar?
Un 0.0% dels punts tals que 0.9<T<Tmàx ha caigut en un radi de 10 m del centroide
Zona de segmentació de temperatura més alta de la descàrrega: Flama
Distància entre el centroide de la descàrrega i el de la zona Flama: 8 m,
queda, doncs, dins del radi de 10 m del centroide de la descàrrega.
Qüestió 5

Quin percentatge de la longitud de la descàrrega ha resultat més efectiu?
Qüestió 6

S'ha solapat correctament la descàrrega en estudi amb la descàrrega precedent?
Aquesta descàrrega està precedida per la descàrrega Drop 1
Temps (min:seg) entre els tdi de les descàrregues: 01:23
Àrea solapada: 0 m²
L'àrea solapada representa un 0% de la descàrrega Drop 1
L'àrea solapada representa un 0% de la descàrrega Drop 2
Vpropagació entre els tdi de les descàrregues a la zona Drop 2: 48 m/min

Qüestió 7

Hi ha zones sense cobrir dins la descàrrega?

Superfície sense cobrir: 36 m² (un 0.77 % de la superfície de la descàrrega)
Gradient de temperatura entre Td i Tdf [%]
INFORME D'AVALUACIÓ DE DESCÀRREGA AÈRIA

Aquest informe està orientat a facilitar la resposta de les qüestions clau proposades a Desenvolupament d'una metodologia basada en termografia infrarroja per a l'anàlisi quantitativa de productes químics en l'extinció d'incendis forestals.

Contents

- Detalls de la descàrrega avaluada
- Qüestió 1
- Qüestió 2
- Qüestió 3
- Qüestió 4
- Qüestió 5
- Qüestió 6
- Qüestió 7

Informe generat el dia 27/10/2009 a les 17:30:55

Detalls de la descàrrega avaluada

Id. Experiment: AS2E
Id. Descàrrega: Drop 3

Qüestió 1

La descàrrega ha assolit a l'objectiu desitjat? (Foc puntual, cap, flanc, assistència dels mitjans terrestres).

Hi ha 0 focs secundaris a l'instant tdi.
Qüestió 2

L'objectiu triat era el més adequat? (S'ha fixat com a objectiu la zona més activa del perímetre de l'incendi o del flanc triat?)

Temperatura aparent entre d1 i d2 respecte Tmin i Tmax de tot el perímetre:
Quartil 1 (Percentil 2)

Velocitat de propagació entre d1 i d2 respecte Vp.min i Vp.màx de tot el perímetre:
Quartil 1 (Percentil 3)
Qüestió 3

La descàrrega ha assolit la millor orientació en relació amb el vent, els paràmetres de vol i la direcció de propagació del foc?

Àrea de la descàrrega: 5328 m²
Dosi efectiva: 0.60 l/m²
Perímetre de la descàrrega: 436 m
Centroide: (485753 Easting, 6042523 Northing)
Longitud màxima: 125 m
Amplada màxima: 63 m (Situada a Inf·Lmax)
Amplada mitjana del tram 0.25·Lmax – 0.75·Lmax: 48.7 m (Desviació estàndard: 6.48 m)
Longitud L': 121 m
Direcció de Lmax: 260º respecte el nord
Direcció de vol: 261º respecte el nord
Direcció del vent: 37º respecte el nord
Direcció de propagació: NaNº respecte el nord
Qüestió 4
La descàrrega ha assolit el tram pretès del perímetre? Hi ha zones ja cremades o combustible sense cremar?
Un 0.0\% dels punts tals que 0.9<T<T_{\text{màx}} ha caigut en un radi de 10 m del centroide.

Zona de segmentació de temperatura més alta de la descàrrega: Residual.

Distància entre el centroide de la descàrrega i el de la zona Residual: 20 m, queda, doncs, fora del radi de 10 m del centroide de la descàrrega.
**Qüestió 5**

Quin percentatge de la longitud de la descàrrega ha resultat més efectiu?
Qüestió 6

S'ha solapat correctament la descàrrega en estudi amb la descàrrega precedent?
Aquesta descàrrega no té cap descàrrega precedent.

Qüestió 7

Hi ha zones sense cobrir dins la descàrrega?

Superfície sense cobrir: 18 m² (un 0.34 % de la superfície de la descàrrega)
INFORME D'AVALUACIÓ DE DESCÀRREGA AÈRIA

Aquest informe està orientat a facilitar la resposta de les qüestions clau proposades a Desenvolupament d'una metodologia basada en termografia infraroja per a l'anàlisi quantitativa de productes químics en l'extinció d'incendis forestals.

Contents

- Detalls de la descàrrega avaluada
- Qüestió 1
- Qüestió 2
- Qüestió 3
- Qüestió 4
- Qüestió 5
- Qüestió 6
- Qüestió 7

Informe generat el dia 27/10/2009 a les 17:32:09

Detalls de la descàrrega avaluada

Id. Experiment: AS2E
Id. Descàrrega: Drop 4

Qüestió 1

La descàrrega ha assolit a l'objectiu desitjat? (Foc puntual, cap, flanc, assistència dels mitjans terrestres).

Hi ha 0 focs secundaris a l'instant tdi.
Qüestió 2

L'objectiu triat era el més adequat? (S'ha fixat com a objectiu la zona més activa del perímetre de l'incendi o del flanc triat?)

Temperatura aparent entre d1 i d2 respecte Tmin i Tmàx de tot el perímetre:
Quartil 1 (Percentil 8)

Velocitat de propagació entre d1 i d2 respecte Vp.min i Vp.màx de tot el perímetre:
Quartil 1 (Percentil 3)
Qüestió 3

La descàrrega ha assolit la millor orientació en relació amb el vent, els paràmetres de vol i la direcció de propagació del foc?

Àrea de la descàrrega: 6363 m²
Dosi efectiva: 0.50 l/m²
Perímetre de la descàrrega: 500 m
Centroide: (485382 Easting, 6042317 Northing)
Longitud màxima: 158 m
Amplada màxima: 69 m (Situada a 0.12·Lmax)
Amplada mitjana del tram 0.25·Lmax - 0.75·Lmax: 40.6 m (Desviació estàndard: 8.38 m)
Longitud L': 151 m
Direcció de Lmax: 33º respecte el nord
Direcció de vol: 41º respecte el nord
Direcció del vent: 37º respecte el nord
Direcció de propagació: -13º respecte el nord
Qüestió 4

La descàrrega ha assolit el tram pretès del perímetre? Hi ha zones ja cremades o combustible sense cremar?
Un 0.0% dels punts tals que 0.9<T<Tmàx ha caigut en un radi de 10 m del centroide. Zona de segmentació de temperatura més alta de la descàrrega: Incandescent. Distància entre el centroide de la descàrrega i el de la zona Incandescent: 56 m, queda, doncs, fora del radi de 10 m del centroide de la descàrrega.
Qüestió 5

Quin percentatge de la longitud de la descàrrega ha resultat més efectiu?
Qüestió 6

S'ha solapat correctament la descàrrega en estudi amb la descàrrega precedent?
Aquesta descàrrega no té cap descàrrega precedent.

**Qüestió 7**

Hi ha zones sense cobrir dins la descàrrega?

La descàrrega ha quedat coberta completament.
INFORME D'AVALUACIÓ DE DESCÀRREGA AÈRIA

Aquest informe està orientat a facilitar la resposta de les qüestions clau proposades a
Desenvolupament d'una metodologia basada en termografia infraroja per a l'anàlisi quantitativa de
productes químics en l'extinció d'incendis forestals.

Contents

- Detalls de la descàrrega avaluada
- Qüestió 1
- Qüestió 2
- Qüestió 3
- Qüestió 4
- Qüestió 5
- Qüestió 6
- Qüestió 7

Informe generat el dia 27/10/2009 a les 17:15:03

Detalls de la descàrrega avaluada

Id. Experiment: AS2E
Id. Descàrrega: Drop 5

Qüestió 1

La descàrrega ha assolit a l'objectiu desitjat? (Foc puntual, cap, flanc, assistència dels mitjans
terrestres).

Hi ha 0 focs secundaris a l'instant tdi.
Qüestió 2

L'objectiu triat era el més adequat? (S'ha fixat com a objectiu la zona més activa del perímetre de l'incendi o del flanc triat?)

Temperatura aparent entre d1 i d2 respecte Tmin i Tmàx de tot el perímetre:
Quartil 1 (Percentil 10)

Velocitat de propagació entre d1 i d2 respecte Vp.min i Vp.màx de tot el perímetre:
Quartil 1 (Percentil 1)
Qüestió 3

La descàrrega ha assolit la millor orientació en relació amb el vent, els paràmetres de vol i la direcció de propagació del foc?

Àrea de la descàrrega: 4815 m²
Dosi efectiva: 0.66 l/m²
Perímetre de la descàrrega: 437 m
Centroide: (485638 Easting, 6042532 Northing)
Longitud màxima: 122 m
Amplada màxima: 63 m (Situada a Inf·Lmax)
Amplada mitjana del tram 0.25·Lmax - 0.75·Lmax: 48.6 m (Desviació estàndard: 7.20 m)
Longitud L': 98 m
Direcció de Lmax: 280° respecte el nord
Direcció de vol: 262° respecte el nord
Direcció del vent: 13° respecte el nord
Direcció de propagació: NaN° respecte el nord
Qüestió 4

La descàrrega ha assolit el tram pretès del perímetre? Hi ha zones ja cremades o combustible sense cremar?
Un 0.0% dels punts tals que 0.9<T谭màx ha caigut en un radi de 10 m del centroide
Zona de segmentació de temperatura més alta de la descàrrega: Residual
Distància entre el centroide de la descàrrega i el de la zona Residual: 26 m,
queda, doncs, fora del radi de 10 m del centroide de la descàrrega.
Qüestió 5
Quin percentatge de la longitud de la descàrrega ha resultat més efectiu?
Qüestió 6

S'ha solapat correctament la descàrrega en estudi amb la descàrrega precedent?
Aquesta descàrrega està precedida per la descàrrega Drop 3
Temps (min:seg) entre els tdi de les descàrregues: 09:34
Àrea solapada: 45 m²
L'àrea solapada representa un 1% de la descàrrega Drop 3
L'àrea solapada representa un 1% de la descàrrega Drop 5
V'propagació entre els tdi de les descàrregues a la zona Drop 5: 3 m/min

Qüestió 7

Hi ha zones sense cobrir dins la descàrrega?

Superfície sense cobrir: 126 m² (un 2.62 % de la superfície de la descàrrega)
Gradient de temperatura entre Td i Tdf [%]

Published with MATLAB 7.8
INFORME D'AVALUACIÓ DE DESCÀRREGA AÈRIA

Aquest informe està orientat a facilitar la resposta de les qüestions clau proposades a Desenvolupament d'una metodologia basada en termografia infraroja per a l'anàlisi quantitativa de productes químics en l'extinció d'incendis forestals.

Contents

- Detalls de la descàrrega avaluada
- Qüestió 1
- Qüestió 2
- Qüestió 3
- Qüestió 4
- Qüestió 5
- Qüestió 6
- Qüestió 7

Informe generat el dia 27/10/2009 a les 17:24:60

Detalls de la descàrrega avaluada

Id. Experiment: AS2E
Id. Descàrrega: Drop 6

Qüestió 1

La descàrrega ha assolit a l'objectiu desitjat? (Foc puntual, cap, flanc, assistència dels mitjans terrestres).

Hi ha 0 focs secundaris a l'instant tdi.
Qüestió 2

L'objectiu triat era el més adequat? (S'ha fixat com a objectiu la zona més activa del perímetre de l'incendi o del flanc triat?)

Temperatura aparent entre d1 i d2 respecte Tmin i Tmax de tot el perímetre:
Quartil 1 (Percentil 5)

Velocitat de propagació entre d1 i d2 respecte Vp.min i Vp.màx de tot el perímetre:
Quartil 1 (Percentil 0)
Qüestió 3

La descàrrega ha assolit la millor orientació en relació amb el vent, els paràmetres de vol i la direcció de propagació del foc?

Àrea de la descàrrega: 5445 m²
Dosi efectiva: 0.59 l/m²
Perímetre de la descàrrega: 495 m
Centoide: (485472 Easting, 6042247 Northing)
Longitud màxima: 170 m
Amplada màxima: 48 m (Situada a 0.55·Lmax)
Amplada mitjana del tram 0.25·Lmax – 0.75·Lmax: 38.9 m (Desviació estàndard: 5.75 m)
Longitud L’: 165 m
Direcció de Lmax: 42º respecte el nord
Direcció de vol: 47º respecte el nord
Direcció del vent: 13º respecte el nord
Direcció de propagació: -25º respecte el nord
Qüestió 4

La descàrrega ha assolit el tram pretès del perímetre? Hi ha zones ja cremades o combustible sense cremar?
Un 0.0% dels punts tals que 0.9<T<Tmàx ha caigut en un radi de 10 m del centroide.

Zona de segmentació de temperatura més alta de la descàrrega: Residual.

Distància entre el centroide de la descàrrega i el de la zona Residual: 20 m, queda, doncs, fora del radi de 10 m del centroide de la descàrrega.
Qüestió 5
Quin percentatge de la longitud de la descàrrega ha resultat més efectiu?
Qüestió 6

S'ha solapat correctament la descàrrega en estudi amb la descàrrega precedent?
Aquesta descàrrega està precedida per la descàrrega Drop 4
Temps (min:seg) entre els tdi de les descàrregues: 09:56
Àrea solapada: 837 m²
L'àrea solapada representa un 13% de la descàrrega Drop 4
L'àrea solapada representa un 15% de la descàrrega Drop 6

V'propagació entre els tdi de les descàrregues a la zona Drop 6: 2 m/min

Qüestió 7

Hi ha zones sense cobrir dins la descàrrega?

Superfície sense cobrir: 153 m² (un 2.81% de la superfície de la descàrrega)
Gradient de temperatura entre Tdi i Tdf [%]

Published with MATLAB® 7.8
Aquest informe està orientat a facilitar la resposta de les qüestions clau proposades a Desenvolupament d'una metodologia basada en termografia infraroja per a l'anàlisi quantitativa de productes químics en l'extinció d'incendis forestals.

Contents

- Details de la descàrrega avaluada
- Qüestió 1
- Qüestió 2
- Qüestió 3
- Qüestió 4
- Qüestió 5
- Qüestió 6
- Qüestió 7

Informe generat el dia 27/10/2009 a les 17:26:54

Details de la descàrrega avaluada

Id. Experiment: AS2E
Id. Descàrrega: Drop 7

Qüestió 1

La descàrrega ha assolit a l'objectiu desitjat? (Foc puntual, cap, flanc, assistència dels mitjans terrestres).

Hi ha 0 focs secundaris a l'instant tdi.
Qüestió 2

L'objectiu triat era el més adequat? (S'ha fixat com a objectiu la zona més activa del perímetre de l'incendi o del flanc triat?)

Temperatura aparent entre d1 i d2 respecte Tmin i Tmàx de tot el perímetre:
Quartil 1 (Percentil 20)

Velocitat de propagació entre d1 i d2 respecte Vp.min i Vp.màx de tot el perímetre:
Quartil 1 (Percentil 0)
**Qüestió 3**

La descàrrega ha assolit la millor orientació en relació amb el vent, els paràmetres de vol i la direcció de propagació del foc?

Àrea de la descàrrega: 5409 m²
Dosi efectiva: 0.59 l/m²
Perímetre de la descàrrega: 428 m
Centroide: (485601 Easting, 6042553 Northing)
Longitud màxima: 124 m
Amplada màxima: 69 m (Situada a Inf·Lmax)
Amplada mitjana del tram 0.25·Lmax - 0.75·Lmax: 51.3 m (Desviació estàndard: 15.21 m)
Longitud L': 123 m
Direcció de Lmax: 278º respecte el nord
Direcció de vol: 274º respecte el nord
Direcció del vent: 34º respecte el nord
Direcció de propagació: NaNº respecte el nord
Qüestió 4

La descàrrega ha assolit el tram pretès del perímetre? Hi ha zones ja cremades o combustible sense cremar?
Un 0.0% dels punts tals que 0.9<T<T_{\text{màx}} ha caigut en un radi de 10 m del centroide.
Zona de segmentació de temperatura més alta de la descàrrega: Residual.
Distància entre el centroide de la descàrrega i el de la zona Residual: 30 m, queda, doncs, fora del radi de 10 m del centroide de la descàrrega.
Qüestió 5

Quin percentatge de la longitud de la descàrrega ha resultat més efectiu?
Qüestió 6

S'ha solapat correctament la descàrrega en estudi amb la descàrrega precedent?
Aquesta descàrrega està precedida per la descàrrega Drop 5
Temps (min:seg) entre els tdi de les descàrregues: 11:02
Àrea solapada: 1602 m²
L'àrea solapada representa un 33% de la descàrrega Drop 5
L'àrea solapada representa un 30% de la descàrrega Drop 7
V'propagació entre els tdi de les descàrregues a la zona Drop 7: 0 m/min

Qüestió 7

Hi ha zones sense cobrir dins la descàrrega?

Superfície sense cobrir: 63 m² (un 1.16 % de la superfície de la descàrrega)
Gradient de temperatura entre Td i Tdf [%]