UNIVERSITÉ ANTONINE
,Faculté d'Ingénieurs En Informatique
Multimédia, Rèseaux & Télécommunications

Fire detection extracted from webcam

Prepared by: SALLOUM HASSAN

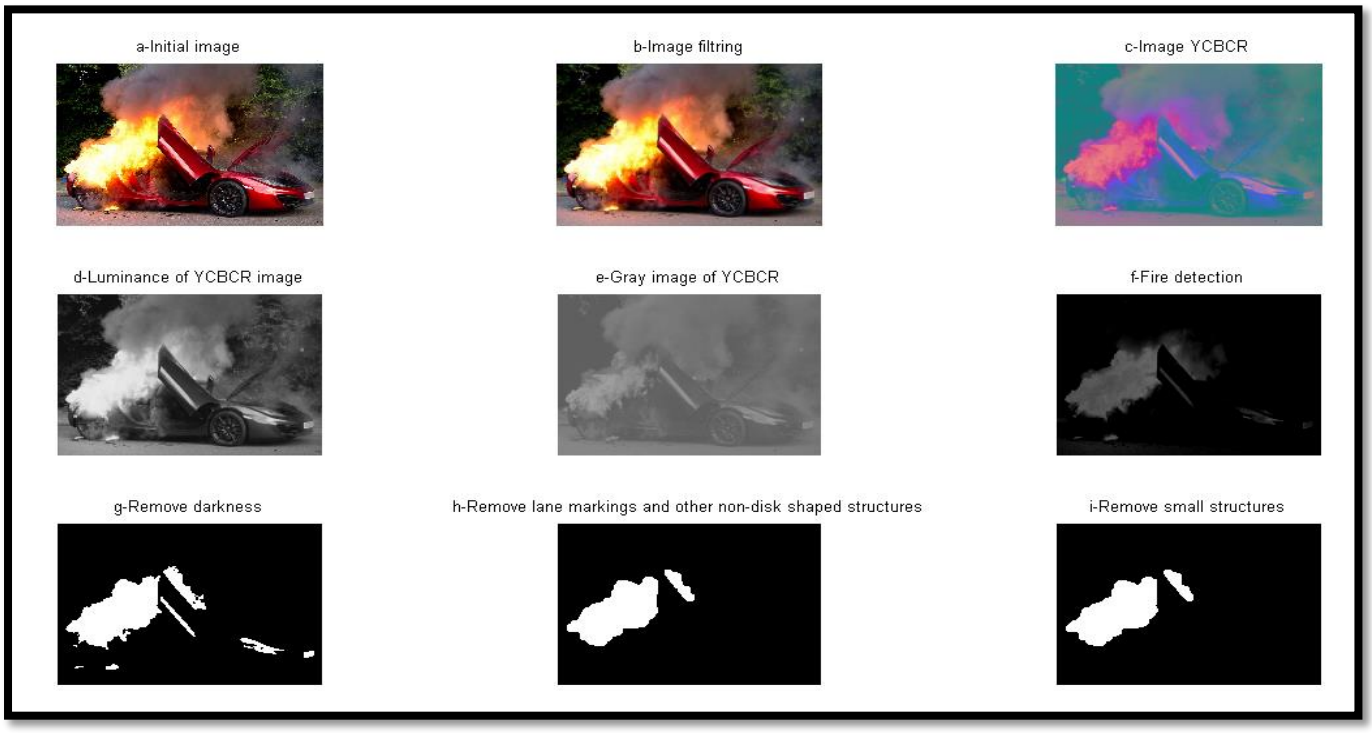
Baabda/ Novomber 2013

PART I: Capture fire from image

How can we distinct the fire color from images and capturing it :

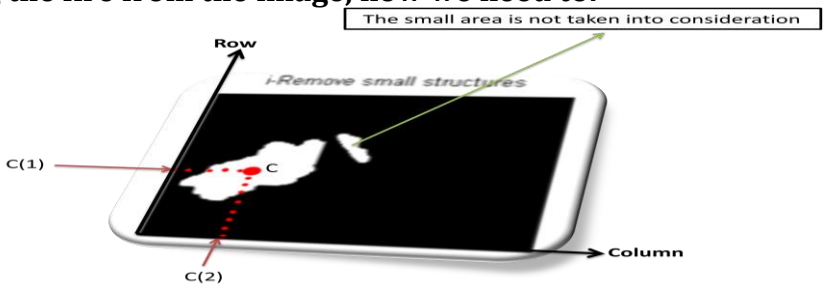
1. Step (a-b-c): distinct the fire color in image by converting the rgb image to YCBCR type
2. Step(d-e-f): substract the fire color in image from the intersection between the gray image (converted from the YCBCR) and the red (converted from the YCBCR)
3. Step(g-h-i): remove all the darkness and all small structures and lane marking and other non-disk shaped structures

a- **RGB = imread('fire2.png');**	**b-** **RGB=imresize(RGB, [360 360]);** **RGB=imfilter(RGB,H);**	**c-** **YCBCR = rgb2ycbcr(RGB);**
d- **im_red=YCBCR(:,:,1);**	**e-** **im_gray=rgb2gray(YCBCR);**	**f-** **im_diff=imsubtract(im_red,im_gray);**
g- **noDarkCars = imextendedmax(im_diff, darkCarValue);**	**h-** **noSmallStructures = imopen(noDarkCars, sedisk);**	**i-** **noSmallStructures = bwareaopen(noSmallStructures, 500);**



PART I: Centroid the fire area

After substracting the fire from the image, now we need to:



1. Get the area and centroid of each remaining object in the image.
2. The object with the largest area is the light-colored fire.
3. Create a copy of the original image and tag the fire by changing the centroid pixel value to red.

taggedFire(:,:,:,k) = singleFrame; // K represented one specific frame in video

stats = regionprops(noSmallStructures, {'Centroid','Area'}) // measures a set of properties for each connected component(object) in image. Properties can be a comma-separated list of strings in this case regionprops computes the 'Area', 'Centroid', measurements

if ~isempty([stats.Area])

areaArray = [stats.Area];

[junk,idx] = max(areaArray);

c = stats(idx).Centroid;

c = floor(fliplr(c));

floor() //rounds the elements of A to the nearest integers less than or equal to A.

B = fliplr(A) //returns A with columns flipped in the left-right direction, that is, about a vertical axis.

c =320.5000 240.5000

fliplr(c)

c=240.5000 320.5000

floor(c)

c = 240 320

width = 2; //for determine the centroid pixel with a high dimension

row = c(1)-width:c(1)+width //row =236 237 238 239 240 241 242 243 244

col = c(2)-width:c(2)+width; //col =316 317 318 319 320 321 322 323 324

taggedFire (row,col,1,k) = 255; //the red color get the high value

taggedFire (row,col,2,k) = 0; //the green color get the low value

taggedFire (row,col,3,k) = 0; //theblue color get the low value

end


Digital Media Converter Pro Trial Version
Please Purchase


Digital Media Converter Pro Trial Version
Please Purchase

PART I: Final code

```
Obj = VideoReader('fire2.avi');
get(Obj);
J= read(Obj, inf);
nframes = get(Obj, 'NumberOfFrames')
taggedFire = zeros([size(J,1) size(J,2) 3 nframes], class(J));
for k = 1:nframes
    darkValue = 50;
    singleFrame = read(Obj, k);
    t=[9 9];
    H = fspecial('Gaussian', t,1);
    ImageG=imfilter(singleFrame,H);
    YCBCR = rgb2ycbcr(ImageG);
    im_red=YCBCR(:,:,1);
    im_gray = rgb2gray(YCBCR);
    I=imsubtract(im_red,im_gray);
    noDark = imextendedmax(I, darkValue);
    sedisk = strel('disk',2);
    noSmallStructures = imopen(noDark, sedisk);
    noSmallStructures = bwareaopen(noSmallStructures, 150);
    taggedFire(:,:,:,k) = singleFrame;
    stats = regionprops(noSmallStructures, {'Centroid','Area'});
    if ~isempty([stats.Area])
        areaArray = [stats.Area];
        [junk,idx] = max(areaArray);
        c = stats(idx).Centroid;
        c = floor(fliplr(c));
        width = 2;
        row = c(1)-width:c(1)+width;
        col = c(2)-width:c(2)+width;
        taggedFire (row,col,1,k) = 255;
        taggedFire (row,col,2,k) = 0;
        taggedFire (row,col,3,k) = 0;
     end

end
        frameRate = get(Obj,'FrameRate');
        implay(taggedFire,frameRate);
```

PART II: Creat video input from webcam

```
vid = videoinput('winvideo', 1, 'MJPG_640x480');
 //Construct a video input object
src = getselectedsource(vid);
//View the properties for the selected video source object.
vid.FramesPerTrigger = 1;
set(vid,'TriggerRepeat', Inf);
// Configure the number of frames to log upon triggering.
vid.ReturnedColorspace = 'rgb';
hVideoOut = vision.VideoPlayer;
hVideoOut.Name  ='Fire detected';
start(vid);
```

Description

obj = videoinput(adaptorname)

Constructs the video input object obj. A video input object represents the connection between MATLAB and a particular image acquisition device.

Adaptorname is a text string that specifies the name of the adaptor used to communicate with the device.

Use the '**IMAQHWINFO**' function to determine the adaptors available on your system.

obj = videoinput(adaptorname,deviceID,format,P1,V1,...)

creates a video input object obj with the specified property values. If an invalid property name or property value is specified, the object is not created.gf

```
>>imaqhwinfo(winvideo)
ans =
InstalledAdaptors: {'matrox'  'winvideo'}
MATLABVersion: '7.12 (R2011a)'
ToolboxName: 'Image Acquisition Toolbox'
ToolboxVersion: '4.1 (R2011a)'
```

PART II: Creat timer

```
c=clock;
display('year month day hour minute seconds')
```

Description

c = clock returns a six-element date vector containing the current date and time in decimal form:[year month day hour minute seconds]

PART II: Creat sound effect when the fire detect

```
y=fix(c)
cf = 2000;              // carrier frequency (Hz)
sf = 22050;             // sample frequency (Hz)
d = 5.0;                // duration (s)
n = sf * d;             // number of samples
s = (1:n) / sf;         // sound data preparation
s = sin(2 * pi * cf * s);  // sinusoidal modulation
sound(s, sf);           //sound presentation
pause(d + 0.5);         // waiting for sound end
```

PART II: Final code

```
vid = videoinput('winvideo', 1, 'MJPG_640x480');
src = getselectedsource(vid);
vid.FramesPerTrigger = 1;
vid.ReturnedColorspace = 'rgb';
set(vid,'TriggerRepeat', Inf);
hVideoOut = vision.VideoPlayer;
hVideoOut.Name  ='Fire detected';
start(vid);
nframes= 300;
for k = 1:nframes
black=0;
white=0;
RGB=getsnapshot(vid); //Acquire and display a single image frame.
t=[9 9];
H = fspecial('Gaussian', t,1);
RGBF=imfilter(RGB,H);
YCBCR = rgb2ycbcr(RGBF);
im_red=YCBCR(:,:,1);
im_gray=rgb2gray(YCBCR);
im_diff=imsubtract(im_red,im_gray);
darkCarValue = 50;
noDarkCars = imextendedmax(im_diff, darkCarValue);
sedisk = strel('disk',10);
noSmallStructures = imopen(noDarkCars, sedisk);
noSmallStructures = bwareaopen(noSmallStructures, 500);
taggedFire= zeros(size(RGB,1),size(RGB,2),3,nframes);
taggedFire(:,:,:,k) = RGB;
 stats = regionprops(noSmallStructures, {'Centroid','Area'});
if ~isempty([stats.Area])
areaArray = [stats.Area]  ;
if(areaArray~=307200)
c=clock;
display('year month day hour minute seconds')
```

```
y=fix(c)
cf = 2000;                // carrier frequency (Hz)
sf = 22050;               // sample frequency (Hz)
d = 5.0;                  //duration (s)
n = sf * d;               // number of samples
s = (1:n) / sf;           // sound data preparation
s = sin(2 * pi * cf * s);   // sinusoidal modulation
sound(s, sf);             // sound presentation
pause(d + 0.5);           // waiting for sound end
end
[junk,idx] = max(areaArray);
x = stats(idx).Centroid
x = floor(fliplr(x))
width = 4;
row = x(1)-width:x(1)+width;
col = x(2)-width:x(2)+width   ;
taggedFire (row,col,1,k) = 255;
taggedFire (row,col,2,k) = 0;
taggedFire (row,col,3,k) = 0;
end
frameRate = get(vid);
step(hVideoOut,[taggedFire(:,:,:,k) RGB]);
end
delete(vid); //Remove video input object from memory.
```