

Additional File 4: Acquisition scheme for alternate camera model (FLIR Systems A320)

The acquisition scripts in this paper are designed to interface with the Mikron M7500 camera. We also developed an acquisition script to work with the FLIR Systems A320 thermal camera. This script worked in the lab, but we did not field deploy the camera and use the script in a field setting. This script performs a low-frame-rate acquisition only, and essentially replaces Scripts L1 and L1b described above. Higher rate acquisitions are capable with this camera but would require substantial modifications to the code. One point to note is that the FLIR ActiveX COM control can be accessed directly within Matlab, unlike the Mikron ActiveX control. This means that there is no need for the Visual BASIC .NET program (Script L1b) for the FLIR A320 acquisition, making the implementation simpler.

The FLIR A320 also has the capability to automatically FTP images out at a set interval (no more frequent than once per minute), thereby avoiding use of the ActiveX controls for image pulling. The FTP settings are managed through the free IR Monitor software.

```
function AcquireFLIRImage01
%function AcquireFLIRImage01
%
%This function acquires a single image (in three different formats) from a
% FLIR A320 using the ActiveX control offered by the FLIR ThermoVision SDK
% software. This program is designed to be run by a scheduler (e.g. Windows
% Scheduled Tasks) to acquire images at a frequency of your
% choosing. Here is a step-by-step outline of what the program does:
%
% 1) Pings camera 10 times to wake it up
% 2) Connects to camera using ActiveX control CAMCTRL.LVCamCtrl.4
% 3) Acquires .IMG image - this is the proprietary FLIR format, which is
% the easiest format for analyses with the FLIR software
% 4) Acquires binary (.bin) image - this is a generic format good for
% long-term archiving, but is harder for quick analyses
% 5) Makes .JPG image for archiving - good for a "quick look" of what's
% going on
% 6) Puts .IMG, .bin, and .JPG files in appropriate date folder (year,
% month, day) - will make folders if necessary
% 7) Does simple temperature measurement (max in image, and temperature at
% three spots), writes this to a running text file
% 8) Makes .jpg file and puts into an html directory, and constantly
% updates this file. This will allow users to have a web page with a
% constantly updated, current image from the camera - like a low-bandwidth
% version of FLIR's IR Monitor software
% 9) Delete any incorrectly acquired files
%10) Automatically exit Matlab
%
%=====
% What YOU need to do:
%=====
% 0) Of course, install FLIR SDK software (which has the ActiveX control
% used in this program), and setup the camera on the network. Use FLIR IR
% Monitor to check that you can connect successfully to the camera.
%
% 1) Make a folder for the current camera, such as C:\FLIR\Cam01
%
% 2) In Part 1, put the image coordinates of three spots that you would
% like to have the temperature monitored at (this can easily be modified by
```

```

% someone comfortable with Matlab to add more points, or make the spots
% boxes)
%
% 3) In Part 2, below, insert the correct IP address, time zone (of the
% clock of the computer doing the acquisition) and folder name
%
% 4) Make a .bat file that will run this Matlab function
% Make a Windows Scheduled Task to run the .bat file
% at a frequency of your choosing (e.g. every 10 minutes).
%
% 5) Setup the html file that will show the current image
%
% 6) Sit back and be amazed at the images that are coming in!
%=====
%
% Oh, and this is important:
% This function is intended for relatively slow acquisition rates - say
% every few minutes. Because this function takes about 30 seconds
% to run, I would not choose your acquisition frequency to be any faster than
% 1 minute.
%
% Matt Patrick
% Hawaiian Volcano Observatory
% US Geological Survey
%
%
% February 20, 2011

%=====
%Part 1: Designate measurement points: MODIFY THESE!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
%=====

%first measurement spot (row, column)
r1=160;
c1=120;
%second measurement spot
r2=100;
c2=100;
%third measurement spot
r3=200;
c3=200;

%=====
%Part 2: Setup IP address and directories: MODIFY THESE!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
%=====

%IP address of camera
IP='192.168.2.62';

%camera name(choose something relatively short)
Cam='Cam01';

%name the time zone used for this computer's clock (i.e. image time stamp)
timezone='HST';

%cd to camera image directory (archive subfolders will be created here)
cc='C:\FLIR\Cam01';
cd(cc);

%=====
%Part 3: Connect to camera and acquire single image
%=====

```

```

%ping camera 10 times before connecting
system(['ping ' IP ' -n 10']);

% Launch the Active-X control!
h = actxcontrol('CAMCTRL.LVCamCtrl.4');

%connect to the camera
a=invoke(h,'Connect',11,0,6,2,IP);

%wait 10 seconds to ensure connection
pause(10);

%following code is commented out, can use it to change camera properties

% d= invoke(h,'SetCameraProperty',91,4)
% d=invoke(h,'SetCameraProperty',91,4)
% d4=invoke(h,'GetCameraProperty',91)
% pause(10)
% d1=invoke(h,'SetCameraProperty',92,1)
% methodsview(h)
% pause(10)
% d4=invoke(h,'GetCameraProperty',92)
% % inspect(h)
% pause(10)

%set acquisition directory for camera
d2=invoke(h,'SetCameraProperty',39,cc)
pause(5)
%acquire .IMG image
e=invoke(h,'DoCameraAction',6)
pause(5)
%if image acquisition failed, try twice more
if e>0
    e=invoke(h,'DoCameraAction',6);
    pause(5)
end
if e>0
    e=invoke(h,'DoCameraAction',6);
end

t1=now;
%acquire one binary image from camera
A = invoke(h,'GetImage',3);
t2=now;
t=(t1+t2)/2; %time of image

%wait 5 seconds
pause(5);

%disconnect from camera
b=invoke(h,'Disconnect');

%wait 2 seconds
pause(2);

%close activex window
close

%save binary image in local folder
year=datestr(now,10);
month=datestr(now,5);
day=datestr(now,7);

```

```

hour=datestr(now,'HH');
mint=datestr(now,'MM');
sec=datestr(now,'SS');
Aname1=[year month day hour mint sec Cam '.bin']; %filename based on time
fid=fopen(Aname1,'w');
c=fwrite(fid,A,'float32'); %saves as 32bit floating point data
fclose(fid);

%=====
%Part 4: Rename and archive .IMG file in correct folder
%=====

%rename IMG file to date-based name
d=dir('*.IMG');
for i=1:length(d)
    year=datestr(d(i).date,10);
    month=datestr(d(i).date,5);
    day=datestr(d(i).date,7);
    hour=datestr(d(i).date,'HH');
    mint=datestr(d(i).date,'MM');
    sec=datestr(d(i).date,'SS');
    Aname2=[year month day hour mint sec Cam '.IMG']; %filename based on time
    if strcmp(d(i).name,Aname2)==0
        movefile(d(i).name,Aname2);
    end
end

%move IMG file to right directory
minute=mint;

d1=dir('*.IMG');
for x=1:length(d1)
    Aname2=d1(x).name;
    year=datestr(d1(x).date,10);
    month=datestr(d1(x).date,5);
    day=datestr(d1(x).date,7);
    hour=datestr(d1(x).date,'HH');
    mint=datestr(d1(x).date,'MM');
    sec=datestr(d1(x).date,'SS');
    d=dir;

    %check if year folder exists
    key=0;
    for j=1:length(d)
        dyear=d(j).name;
        if strcmp(year,dyear)==1 & d(j).isdir==1
            key=1;
        end
    end
    if key==1
        cd(year);
    else mkdir(year)
        cd(year);
    end

    %check if month folder exists
    key=0;
    d=dir;
    for j=1:length(d)
        dmonth=d(j).name;
        if strcmp(month,dmonth)==1 & d(j).isdir==1
            key=1;
        end
    end
end

```

```

end
if key==1
    cd(month);
else mkdir(month)
    cd(month);
end

%check if day folder exists
d=dir;
key=0;
for j=1:length(d)
    dday=d(j).name;
    if strcmp(day,dday)==1 & d(j).isdir==1
        key=1;
    end
end
if key==1
    cd(day);
else mkdir(day)
    cd(day)
end

%uncheck section below if using hour folders:

% %check if hour folder exists
% d=dir;
% key=0;
% for j=1:length(d)
%     j;
%     dhour=d(j).name;
%     if strcmp(hour,dhour)==1 & d(j).isdir==1
%         key=1;
%     end
% end
% if key==1
%     cd(hour);
% else mkdir(hour)
% end

%go back to root directory
cd(cc);
%uncomment line below if using hour folders, and comment out line after that
%movefile(Aname2,[year '/' month '/' day '/' hour '/' Aname2]);
movefile(Aname2,[year '/' month '/' day '/' Aname2]);

B=rot90(A,1); %rotate image upright
B=flipud(B); %flip left to right
B=B-273.15; %convert from K to C
imagesc(B)
set(gca,'xtick',[])
set(gca,'ytick',[])
axis equal
axis tight
colorbar
load iron
colormap(iron)
title(['Cam ' ' datestr(t,0) ' ' timezone])
cd(year)
cd(month)
cd(day)
%cd(hour)
%make jpg image
print('-r150','-djpeg85',[year month day hour mint sec Cam '.jpg']);

```

```

    cd(cc)
end

%=====
%Part 5: If binary image was not acquired correctly, try again
%=====

if size(A,1)<2
    % Launch the Active-X control!
    h = actxcontrol('CAMCTRL.LVCamCtrl.4');
    %connect to the camera
    a=invoke(h, 'Connect', 11, 0, 6, 2, IP);

    %wait 10 seconds to ensure connection
    pause(10);

    t1=now;
    %acquire one image from camera
    A = invoke(h, 'GetImage', 3);
    t2=now;
    t=(t1+t2)/2; %time of image
    %wait 2 seconds
    pause(2);
    %disconnect from camera
    b=invoke(h, 'Disconnect');
    %wait 2 seconds
    pause(2);
    %close activex window
    close
    %save image in local folder
    year=datestr(now,10);
    month=datestr(now,5);
    day=datestr(now,7);
    hour=datestr(now, 'HH');
    mint=datestr(now, 'MM');
    sec=datestr(now, 'SS');
    Aname1=[year month day hour mint sec Cam '.bin']; %filename based on time
    fid=fopen(Aname1, 'w');
    c=fwrite(fid, A, 'float32'); %saves as 32bit floating point data
    fclose(fid);
end

%=====
%Part 6: If the binary image was acquired correctly then do analysis and
%put the image in the right directory (if not, just give up and exit)
%=====

if size(A,1)>3
    cd(cc)
    %=====
    %Part 6a: Make simple temperature measurements
    %=====

    %measure maximum temperature in image
    Tmax=max(max(A))

    %measure temperatures at three spots specified in Part 1
    T1=A(r1,c1)
    T2=A(r2,c2)
    T3=A(r3,c3)

    %=====
    %Part 6b: now put the image file in the right date directory. If

```

```

%directory doesn't exist, make it
%=====
minute=mint;

d=dir;

%check if year folder exists
key=0;
for j=1:length(d)
    dyear=d(j).name;
    if strcmp(year,dyear)==1 & d(j).isdir==1
        key=1;
    end
end
if key==1
    cd(year);
else mkdir(year)
    cd(year);
end

%check if month folder exists
key=0;
d=dir;
for j=1:length(d)
    dmonth=d(j).name;
    if strcmp(month,dmonth)==1 & d(j).isdir==1
        key=1;
    end
end
if key==1
    cd(month);
else mkdir(month)
    cd(month);
end

%check if day folder exists
d=dir;
key=0;
for j=1:length(d)
    dday=d(j).name;
    if strcmp(day,dday)==1 & d(j).isdir==1
        key=1;
    end
end
if key==1
    cd(day);
else mkdir(day)
    cd(day)
end

%uncheck section below if using hour folders:

% %check if hour folder exists
% d=dir;
% key=0;
% for j=1:length(d)
%     j;
%     dhour=d(j).name;
%     if strcmp(hour,dhour)==1 & d(j).isdir==1
%         key=1;
%     end
% end
% if key==1

```

```

%     cd(hour);
% else mkdir(hour)
% end

%go back to root directory
cd(cc);
%uncomment line below if using hour folders, and comment out line after that
%movefile(Aname1,[year '/' month '/' day '/' hour '/' Aname1]);
movefile(Aname1,[year '/' month '/' day '/' Aname1]);

else exit
end

%=====
%Part 7: export image to jpg for easy html viewing
%=====

A=rot90(A,1); %rotate image upright
A=flipud(A); %flip left to right
A=A-273.15; %convert from K to C
imagesc(A)
set(gca,'xtick',[])
set(gca,'ytick',[])
axis equal
axis tight
colorbar
colormap(iron)
title([Cam ' ' datestr(t,0) ' ' timezone])
cd(cc)
%check if html folder exists
d=dir;
key=0;
for j=1:length(d)
    dday=d(j).name;
    if strcmp('html',dday)==1 & d(j).isdir==1
        key=1;
    end
end
if key==1
    cd('html');
else mkdir('html')
    cd('html')
end
print('-r300','-djpeg85',[Cam '_CurrentImage.jpg']); %make jpg image

%=====
%Part 8: write temperature measurements to a text file for simple analysis
%=====
cd(cc)
fid=fopen([Cam '_TempData.csv'],'at+');
k=273.15;
fileline=[datestr(t,0) ', ' num2str(Tmax-k) ', ' num2str(T1-k) ', ' ...
    num2str(T2-k) ', ' num2str(T3-k)];
count=fprintf(fid,'%s\n',fileline);
st=fclose(fid);

%=====
%Part 9: delete incorrectly acquired binary images
%=====
cd(cc)
d=dir('*.bin');
for i=1:length(d)
    if d(i).bytes<20

```



```
        delete(d(i).name);  
    end  
end  
exit %exit Matlab upon completion
```