Explaining The Code

Nikhil Nandyala
AKS Teja

Using the Installation pdf install all the libraries required to run this code.TO understand the code better you need to understand our project algorithm and its inner workings.

To run the code you need to setup an ip camera and define the no of states i.e. the presets you set in your camera.

If your camera if different than wanscam go to line 157 to change the url required to get images from your current ip camera. Go to line no 220 and 291 and change the url respectively.

Directory Structure :

Probabilistic PTZ Modeling

---ZoomWebsite
-------camera.py
-------main.py
-------Templates
-------images
-------userinput.txt
---main.cpp
---motion.cpp
---motion.h
---haarcascades
---deploy.prototxt (DNN)
---res10_300x300_ssd_iter_140000.caffemodel(DNN)
--etc

The Code contains mainly four files

1) main.cpp

It contains the complete algorithmic steps

2)Motion.cpp

It contains all the functions used in main.cpp

3)Motion.h

It defines all the functions present in Motion

4)Make file

If you install the libraries using the installation manual you will be able to run makefile easily without any changes.

After installing the librabies go to the makefile directory and $ make to complile your code.

To run your code
./main a b c d

a= No of states you are setting in your camera.

b=display time i.e the number of frames the algorithm has to display the video i.e zooming and tracking time

c=Ip Address of your camera

d= How much time you want your framework to run.

main.cpp is clearly explained line to line using comments.

But the overall view is

Global variables(line 5 -27)

Main function

Variables and detectors initialization (41-92)

While (it runs for running time you give as input) (167-507) :

Gets the current frame from url and altering the frame to fit our detectors features (171 -192)

If frame_counter is between x, y (198-374) :
means it is in analysing phase i.e calculating sensitivity it is only for 5 frames

Initializing phase(203-235) only enters this phase for first 1 min of the complete algorithm running time

When frame_counter==x (beginning of the analysing phase) :
The state selection is done and sleeps for transition or it checks user input if any (237-302)

The frames are analysed using face,Motion etc and sensitivity map is calculated (303-317).

When frame_counter==y(ending of the analysing phase) :
Based on the sensitivity calculated State values are updated and roi is calculated(331-373)

Else (379-489) :
Means it is zooming and tracking the above calculated roi of the currentstate

Checking for motion in roi , so if tracking(382-407)
Tracking if motion in roi (410-432)
Zooming part(432 -474)
In the intilase frames of the zoom and track phase it is zoom in.(gradually) (437-447)
In the middle frames of the zoom and track phase it is stay in the zoomed area. (449-461)
In the end frames of zoom and track phase it is zoom out(gradually) to original view (461-474)

Displaying the zoomed and track view frame and storing the frames in this directory.

Motion.cpp contains the following functions

Some of the important and specific functions that one may be need to change to imporve or understand the project are :

Rect FindContourAndGetMaxRect2(Mat frame)

Input : Matrix/frame which is binary (2d)and sinagle layered no rgb.
Output : Rectangle in opencv rectangle format.

Explanation :
When a frame input is given it calculates the contours and returns the rectangle covering the maximum contouring with maximum sum.

Mat FaceDetectByCascade(Mat frame, Mat mask, CascadeClassifier *face_cascade);

Input : input image from the camera, mask , Cascadeclassfier defined earlier.
Output : A 2d matrix with each pixel representing probabilty of face detection

Explanation:
It does facedetectiona and updates the sensitivity map mask with the face detection matrix which is also the matrix we are returning.

Mat BackGroundDetection(Mat frame, Mat mask,Ptr<BackgroundSubtractor> pMog);

Input : input image from the camera, mask , Background Subractor defined earlier.
Output : A 2d binary matrix with each pixel with '1' if motion present otherwise '0'

Explanation:
It does motion detection and updates the sensitivity map mask with the motion detection matrix which is also the matrix we are returning.

Mat PeopleDetectByHOG(Mat frame , Mat mask, HOGDescriptor hog);

Input : input image from the camera, mask , HogDescripot defined earlier.
Output : A 2d matrix with each pixel representing probabilty of Human detection

Explanation:

It does Human detection and updates the sensitivity map mask with the human detection matrix which is also the matrix we are returning.

Rect Rect_AdjustSizeAroundCenter(Rect rect, double width_factor, double height_factor,
int videoWidth = 0, int videoHeight = 0);

Input :

Output : A 2d matrix with each pixel representing probabilty of Human detection

Explanation:

It does Human detection and updates the sensitivity map mask with the human detection matrix which is also the matrix we are returning.

Mat FaceDetectorDNN(Mat frame, Mat mask, dnn::Net net);

Input : image frame, mask i.e sensitivity map that needs to be updated, the dnn intialsed before

Output : A 2d matrix with each pixel representing probabilty of face detection

Explanation:

It does facedetectiona and updates the sensitivity map mask with the face detection matrix which is also the matrix we are returning.

Mat curlImg(const char *img_url, int timeout);

Input : image url where the camera is streaming, timeout is the time we are allowing for the url

Output : returns image matrix from the url given.

void curlptzcam(const char *img_url , int to);

Input : image url

Output : It calls the ptz camera url which in turn call the camera to go the preset state.

int nextStateGenerator(int A[]);

Input : Sensitive score of each state array

Output : It calculates the next state on the sensitivity array.

Website :

If you want to use the website run the python code named Camera.py from Szoomwebsite folder.

Install flask and run main.py in the SZoom website.

Open browser and run localhost:5000 to get the website.

Uncomment 484 line in main.cpp
And you have to replace the file where the replace image path in line 484 of main.cpp
And you have the replace the file where the replace image path in line no line 7 of camera.py to

images folder in the downloaded Zoom website.

And

If you want to use user input feature in line number 255 change that addres to userinput.txt in zoomwebsite folder.