Project on video segmentation

Mentor – S. Mukhopadhayay

1. Rajesh kumar sinha 2013je0333
2. Ashish kumar 2013je0606
3. Parichaya walia 2013je0336
4. Harshit sinha
5. Ajit singh
6. Maheswara Reddy Chennuru 2013JE0889

Introduction –

This project is based on the video segmentation.

This project has been done in c/c++ using the open cv libraries.

OpenCV is released under a BSD license and hence it’s free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform. Adopted all around the world, OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 9 million. Usage ranges from interactive art, to mines inspection, stitching [maps on](http://opencv.org/) the web or through advanced robotics.

This project includes two programs done in c++ using opencv.

1. This program take the input of a video through the command line and it segments the video into colored frames and saves it into the computer’s harddisk.

Below is the code for the program.

/\*

\* gray.cpp

\*

\* Created on: 15-Oct-2014

\* Author: rajesh

\*/

**#include** <iostream>

**#include** <stdio.h>

**#include** <opencv2/opencv.hpp>

**int** **main** (**int** argc, **char** \*\*argv)

{

**if** (argc !=2)

{

std::cout << "USE: " << argv[0] << " <video-filename>" << std::**endl**;

**return** 1;

}

//Open the video that you pass from the command line

cv::VideoCapture cap(argv[1]);

**if** (!cap.**isOpened**())

{

std::cerr << "ERROR: Could not open video " << argv[1] << std::**endl**;

**return** 1;

}

**int** frame\_count = 0;

**bool** should\_stop = **false**;

**while**(!should\_stop)

{

cv::Mat frame;

cap >> frame; //get a new frame from the video

**if** (frame.**empty**())

{

should\_stop = **true**; //we arrived to the end of the video

**continue**;

}

**char** filename[128];

**sprintf**(filename, "frame\_%06d.jpg", frame\_count);

cv::**imwrite**(filename, frame);

frame\_count++;

}

**return** 0;

}

output of the program :- here the input is a file wild.mp4 which is segmentated into colored frames. Some of the images are given below.





2. This program take the input of a video through the command line and it segments the video into grayscale frames and saves it into the computer’s harddisk.

Below is the code for the program.

/\*

\* converting video to grayscale.cpp

\*

\* Created on: 15-Oct-2014

\* Author: rajesh

\*/

#include<cv.h>

#include<highgui.h>

**int** g\_slider\_position = 0;

CvCapture\* g\_capture = NULL;

**int** pos;

**void** onTrackbarSlide(**int** pos)

{

cvSetCaptureProperty(g\_capture,CV\_CAP\_PROP\_POS\_FRAMES,pos);

}

IplImage\* convert\_grayscale( IplImage\* img )

{

**for**( **int** y=0; y<img->height; y++ )

{

**int** ptr2;

uchar\* ptr = (uchar\*) (img->imageData + y \* img->widthStep);

**for**( **int** x=0; x<img->width; x++ xxxx)

{

ptr2=(ptr[3\*x+1]+ptr[3\*x+2]+ptr[3\*x+3])/3;

ptr[3\*x+1] = ptr2;

ptr[3\*x+2] = ptr2;

ptr[3\*x+3] = ptr2;

}

}

**return** img;

}

**int** main()

{

cvNamedWindow( "black and white", CV\_WINDOW\_AUTOSIZE );

g\_capture = cvCreateFileCapture("harry.mp4");

**int** frames = (**int**) cvGetCaptureProperty(g\_capture,CV\_CAP\_PROP\_FRAME\_COUNT);//counts total no. of frames and saves in frames

IplImage\* frame;

**while**(1)

{

**if**( frames!= 0 )

{

cvCreateTrackbar("Position","Example3",&g\_slider\_position,frames,onTrackbarSlide);//name appearing along the trackbar,window,

}

frame = cvQueryFrame( g\_capture );

**if**( !frame ) **break**;

g\_slider\_position++;

frame=convert\_grayscale(frame);

cvShowImage( "Example3", frame );

**char** c = cvWaitKey(3);

//cvSaveImage("out.jpg",frame);

**if**( c =='x') **break**;

}

cvReleaseCapture( &g\_capture );

cvDestroyWindow( "Example3" );

**return**(0);

}

output of the program -

this program has taken the input of a file “harry.mp4” and it has segmentated it into grayscale frames.

A description...

A description...