

Mean Shift Tracking

P119404

September 24, 2013

Contents

Mean Shift Tracking	3
0.1 Introduction	3
0.2 Mean shift	3
0.3 Using Similarity for tracking	4
0.3.1 Implementation	5
0.4 Code	7

Mean Shift Tracking

0.1 Introduction

In the article we will look at the application of Mean Shift Tracking for color based tracking.

0.2 Mean shift

The object model used in mean shift tracking is color probability distribution.

Now we have a object model, given an image we can compute the likelihood image. Each pixel in likelihood image represents the likelihood that pixel belongs to the object model/histogram.

$$L_u(y_i) = p_u * \delta(b(y_i) - u)$$

This likelihood image assigns to each pixel a similarity measure wrt

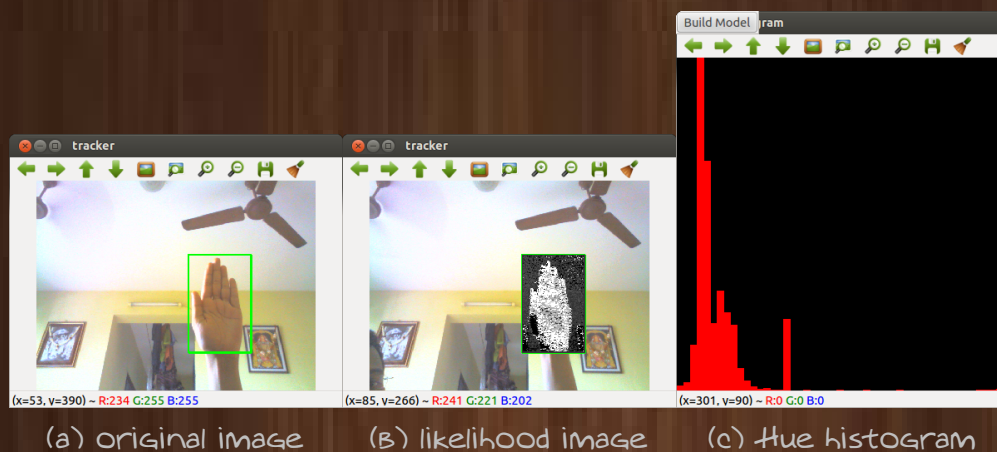


Figure 1: Object model

object model.

It is reasonable to assume that the region in which the highest similarity measure or highest density is observed is a good estimate of object location.

```

1 //building the object appearance model
2 void ocvmeanShift::buildModel(Mat image,Rect rect)
3 {
4     //input region of interest
5     region=rect;
6     //center of region is current location estimate
7     p.x=region.x+region.width/2;
8     p.y=region.y+region.height/2;
9     //extract ROI
10    Mat roi=image(rect);
11    //compute the histogram ,h is object of type Histogram
12    h.BuildHistogram(roi);
13 }
14
15 //call to compute the likelihood after computing the model
16 Mat sim=h.likelihoodImage(image);

```

Thus if we consider a small window and move towards the mean value ie along the mean shift vector we should eventually reach the region of maximum similarity.

The likelihood surface is not smooth,we can give it properties of smoothness using kernel density estimation.

Now we can find the modes of the similarity surface using standard mean shift algorithm.

Let us assume that current estimate of mode of function is at y . Thus we consider a small rectangular window about y ,compute the mean shift vector and take a small step along the mean shift vector.

In principle this should enable to find the local maximum. Similarity surface is discontinuous and to do this we need to perform KDE over entire image consisting of dense grid of points ,which is a expensive operation.(We need to perform convolution at each point with Gaussian with suitably large aperture)

0.3 Using Similarity for tracking

The concept of similarity surface can be made useful in tracking application.

Let us consider a small region of interest about a present location y , we can compute the similarity score about this region, perform KDE on this small region, obtain a similarity surface and compute the mean shift vector.

If object is not present in region, similarity surface will be flat and mean shift vector will be zero.

If there is object present in some part of region, it will correspond to modes of similarity surface. The mean shift vector will give us direction to move along.

Now instead of trying to estimate the mode, say we translate the region of interest along direction provided by the mean shift vector. This would typically lead to large portion of object being visible and would expose the region of global similarity surface where a large maximum would lie.

This is the basis of mean shift tracking, keep on translating the region of interest, till we reach local maximum of similarity surface.

For tracking applications, since fast computation is required, we can consider a rectangular window with Bandwidth equal to that of the region of interest. The present location of point point is the center of the rectangular region.

0.3.1 Implementation

```

1 //compute the likelihood
2 Mat sim=h.likeyhoodImage(image);
3 //perform iteratively till convergence
4 for(int i=0;i<criteria.maxCount;i++)
5 {
6     //extracting the region of interest
7     Mat roi=sim(region);
8     //compute the moments
9     cv::Moments m;
10    m=cv::moments(roi,false);
11
12    //threshold m00 which is weighted mean,
13    //exit since no similar pixels present
14    if(fabs(m.m00)<region.width*region.height*0.05)
15        break;

```

```

16
17 //computing the mean values
18 int x=cvRound(m.m10/m.m00);
19 int y=cvRound(m.m01/m.m00);
20
21 //computing the mean shift
22 int dx=region.width/2-x;
23 int dy=region.height/2-y;
24
25 //displacement from current position
26 int nx=p.x-dx;
27 int ny=p.y-dy;
28
29 //bounday of the image
30 if(nx-region.width/2<=0) nx=region.width/2;
31 if(nx+region.width/2>=image.cols) nx=image.cols-region.width/2-1;
32 if(ny-region.height/2<=0) ny=region.height/2;
33 if(ny+region.height/2>=image.rows) ny=image.rows-region.height/2-1;
34
35 //recalculating the mean shift
36 dx=-nx+p.x;
37 dy=-ny+p.y;
38
39 //checking magnitude of mean shift vector.
40
41 float mag=dx*dx+dy*dy;
42
43 //no change in mean,reached local maxima
44 if(mag<criteria.epsilon*criteria.epsilon)
45     break;
46
47 //updating the position
48 p.x=nx;
49 p.y=ny;
50 //updating the region of interest
51 region.x=p.x-region.width/2;
52 region.y=p.y-region.height/2;
53
54 }

```

A video of mean shift tracking is shown below. A naive object model based on color probability in HS color space using first frame of the video <https://googledrive.com/host/0B-pfqaQBbAAaNkg2bUJvWERmNFU/video00001.ogv>

As will all tracking approaches, the performance heavily depends on the object model. The better we are able to model the object and obtain a likelihood/similarity which does not show high probability for background or other objects in the scene, the more accurate will be the tracking.

0.4 Code

For further image processing application a library consisting of high level interface to opencv will be used. The library is called OpenVision-Library. <https://github.com/pi19404/OpenVision>

The project cmake file is included in the repository. the build will create the library and test files in the bin directory. To run demo program for mean shift run the binary meanShiftTest.

The files for mean shift algorithm are meanshift.cpp and meanshift.hpp <https://github.com/pi19404/OpenVision/tree/master/ImgProc/> repository.

To run the test program :

```
1 meanShift - to run using camera input
2 meanShiftTest {video file name} - to run using a video file
```

Select the region of interest and click the build model button to start tracking.

For video file initially only the first frame is shown. select the ROI in the first frame and then click on build model button to start the tracking.

The button is shown upon clicking on Display properties button on the window.