Registration and Tracking by using Mean Shift Algorithm

Kavya G1, Jharna Majumdar2

1M Tech, Dept of CSE, Nitte Meenakshi Institute of Technology, Yelahanka, Bangalore, Karnataka, India
2Dean R&D, Professor and Head, Dept of CSE (PG), Nitte Meenakshi Institute of Technology, Yelahanka, Bangalore, Karnataka, India

Abstract: Object tracking is the problem of determining (estimating) the positions and other relevant information of moving objects like car, ball or human in a video. There are number of application of tracking surveillance, traffic monitoring, robot vision, animations. Registration is the basic step of tracking. This paper proposes mean shift algorithm which is an iterative method, efficient approach to track a non rigid object. Three different kernels are used for weight distribution, namely Uniform, Gaussian, Epanechnikov kernel. To find the Similarity between the consecutive frames Bhattacharyya coefficient is used. The proposed approach also gives efficient results with different size of searching window.

Keywords: Registration, Object Tracking, Mean Shift, Video, Bhattacharyya Coefficient, kernel density estimation.

1. Introduction

Image Registration is a process of finding the location where optimal matching is obtained by matching a template image called the reference image over the searching region of an input image using a suitable similarity measures. The method although computationally intensive and robust but it is simple, straightforward, requires no priori information about the two images. Image registration has two important requirements, time and accuracy. Tracking is the process of locating object or human over a time using camera [1]. Tracking has two steps target detection and target tracking. Tracking starts by acquiring the initial image and select the object of interest. Acquire next sequence of images locate the target and track the object repeat this until end of frames. The use of object tracking has many vision applications such as automated video surveillance, traffic monitoring, robot vision, animation and security and defense areas. The survey shows huge number of algorithms designed and developed for object tracking [4]. This paper mainly focuses on mean shift algorithm for tracking of object. Mean shift algorithm is an iterative process. The Mean-Shift algorithm tracks by maximizing Bhattacharyya coefficient or minimizing a distance from that point to initial point; h is the kernel radius. Where, nh is the number of points inside the kernel; y0 is the initial point.

2. Basic Mean Shift Analysis

The basic idea of mean shift is to find the maximum density in the distribution. In order to illustrate the how mean shift works; For example take any distribution of points (p1, p2, p3, ..., pn) which are spread on 2D space. Each of the points have x and y coordinates. Start with any region of interest around that initial point. Calculate the mean of all points inside the region. So move the initial point to new point that will become new initial point repeat the procedure until it converges to maximum density area. The criterion for convergence is mean of all points is same as initial point. Mean shift vector always points towards direction of maximum density. The General form of mean shift is represented as,

$$m_{n}(y) = \frac{1}{n_{h}} \sum_{i=0}^{n_{h}} x_{i} - y_{0}$$

Where, n is the total number of points inside region of interest; x0 is the points x1, x2, ..., xnh; y0 is the initial point.

This is equal weighted mean shift calculation (i.e every points have same weight). The Better way to do this by allotting small weights to each point. It can be represented as,

$$m_{s}(y) = \frac{1}{\sum_{i=1}^{n_{h}} w_{i}(y_{0})x_{i}} \sum_{i=1}^{n_{h}} w_{i}(y_{0})x_{i} - y_{0}$$

Where, n is the number of points inside the kernel; y0 is the initial estimate; w is the weights for each point based on the distance from that point to initial point; h is the kernel radius. Weights are determined by using different kernels such as uniform kernel, Gaussian kernel, Epanechnikov kernel. Mean shift is used for finding modes in a set of samples demonstrate underlying probability distribution (pdfs).

A function with finite number of data points x1, x2, ..., xn

$$P(x) = \frac{1}{n} \sum_{i=1}^{n} k(x - x_{i})$$

2.1 Uniform Kernel

The uniform kernel function is 1/2, for values between -1 and -1 and zero outside that range. Here $$w=(x-x_{i})/h$$, where h is the window width and xi are the values of the independent variable in the data, and x is the value of the independent variable for which one seeks an estimate.
2.2 Normal Kernel

The Gaussian kernel is this function: \((2\pi)^{-\frac{1}{2}}\exp(-u^2/2)\). Here \(u=(x-x_i)/h\), where \(h\) is the window width and \(x_i\) are the values of the independent variable in the data, and \(x\) is the value of the independent variable for which one seeks an estimate. Unlike most kernel functions this one is unbounded on \(x\).

\[K_\sigma(x) = \begin{cases} c \| x \| < 1 \\ 0 \text{ otherwise} \end{cases} \]

(4)

\[K_\sigma(x) = c \exp \left( -\frac{1}{2} \| x \|^2 \right) \]

(5)

2.3 Epanechnikov Kernel

The Epanechnikov kernel is this function: \((3/4)(1-u^2)\) for \(-1<u<1\) and zero for \(u\) outside that range. Here \(u=(x-x_i)/h\), where \(h\) is the window width and \(x_i\) are the values of the independent variable in the data, and \(x\) is the value of the scalar independent variable for which one seeks an estimate.

\[K_E(x) = \begin{cases} c \left( 1 - \| x_i \|^2 \right) \| x \| \leq 1 \\ 0 \text{ otherwise} \end{cases} \]

(6)

3 Bhattacharyya Coefficient

Bhattacharyya coefficient is used for finding the similarities between target in current frame and previous frame. If Bhattacharyya coefficient is maximum then minimizes the dissimilarity. In the proposed method, Bhattacharyya distance measure is applied between target models and candidate in the form of probability distributions. It is represented in the form of metric, \(\rho\).

\[\rho(y) = \rho[p(y), q] = \int \sqrt{p(y)q} \, dz \]

(7)

The feature \(z\) representing the color or intensity of the target model is assumed to have a density function \(q_z\), while the target candidate centered at location \(y\) has the feature distributed according to \(p_z(y)\). The problem is then to find the discrete location \(y\) whose associated density \(p_z(y)\) is the most similar to the target density \(q_z\). As mean shift is a vector so the \(p_z(y)\) and \(q_z\) will be in the form of vectors. Bhattacharyya coefficient is measured between two vectors. \(\rho\) is the dot product of two vectors or angle between two vectors i.e. \((\sqrt{p_1}, ..., \sqrt{p_m})\) and \((\sqrt{q_1}, ..., \sqrt{q_m})\). Larger the value of \(\rho\) means good match is found. In order to find the new target location, try to maximize the Bhattacharyya Coefficient.

4. Proposed Method

Mean shift based tracking has four major steps: 1) object detection in the first frame, 2) tracking of the object in subsequent frames, 3) extraction of features such as color or grey level intensity or texture or edges represent them in the form of probability density function and 4) formulating decision based upon similarity measure obtained using Bhattacharyya coefficient [2][5]. This Mean shift algorithm iteratively finds the exact match of moving object in sequence of frames. The Mean-Shift algorithm tracks by maximizing Bhattacharyya between two probability density functions (pdfs) [5] represented by a reference and candidate histograms. Since the histogram distance (or, equivalently, similarity) does not depend on spatial structure of the search window, the method is suitable for deformable and articulated objects. Histogram does not mean its reserved for color property, it also includes texture or shape (edges). Registration has two important requirements time and accuracy [1]. The experiments are carried out to calculates the time requirement and three accuracy measures (pixel difference, pixel cross correlation, discrete similarity measure). The mean shift vector is updated iteratively towards the convergence where the Bhattacharyya coefficient is maximized. Fig 2 shows flow diagram of mean shift tracking process.

![Figure 1: Pdf representation of target and candidate](image-url)
4.1 Target Model
As mean shift suffers from fixed size search window (i.e., fixed object scale). The target is represented by a rectangular region in the frame. In our approach, experiments with different size of searching window which overcome the fixed size search window problem.

Let \( \{x_i^*\}_{i=1}^{n} \) be the pixel locations of the target model, centered at 0. Pixel at location \( x_i \) the index \( b(x_i) \) of the histogram bin corresponding to the color of that pixel. The probability of the color \( u \) in the target model is derived by employing a convex and monotonic decreasing kernel profile \( k \) which assigns a smaller weight to the locations that are farther from the center of the target. The weighting increases the robustness of the estimation, since the peripheral pixels are the least reliable, being often affected by occlusions (clutter) or background. The radius of the kernel profile is taken equal to one. It is represented as,

\[
\hat{p}_u(y) = c \sum_{i=1}^{n} k\left(\frac{y-x_i^*}{h}\right) \delta[b(x_i^*)-u]
\]  

(10)

Where, \( C_h \) is the normalization constant. The radius of the kernel profile determines the number of pixels (i.e., the scale) of the target candidate. By imposing the condition that \( \sum_{i=1}^{n} \hat{p}_u = 1 \) we obtain

\[
C_h = \frac{1}{\left(\sum_{i=1}^{n} k\left(\frac{y-x_i^*}{h}\right)\right)}
\]  

(11)

Note that \( C_h \) does not depend on \( y \), since the pixel locations \( x_i \) are organized in a regular lattice, \( y \) being one of the lattice nodes. Therefore, \( C_h \) can be recalculated for a given kernel and different values of \( h \).

Weights are calculated by this formula,

\[
w_i = \sum_{u=1}^{m} \delta\left(b(x_i^*)-u\right) \sqrt{\frac{\hat{q}_u}{\hat{p}_u(y)}}
\]  

(12)

4.2 Target candidate
Let \( \{x_i^*\}_{i=1}^{nh} \) be the pixel locations of the target candidate, centered at \( y \) in the current frame. Using the same kernel profile \( k \), but with radius \( h \), the probability of the color \( u \) in the target candidate is given by,

\[
\hat{p}_u(y) = c_h \sum_{i=1}^{n} k\left(\frac{y-x_i}{h}\right) \delta[b(x_i)-u]
\]  

Where \( C_h \) is the normalization constant. The radius of the kernel profile determines the number of pixels (i.e., the scale) of the target candidate. By imposing the condition that \( \sum_{i=1}^{n} \hat{p}_u = 1 \) we obtain

\[
C_h = \frac{1}{\left(\sum_{i=1}^{n} k\left(\frac{y-x_i}{h}\right)\right)}
\]

4.3 Algorithm
Tracking of object or human in the input video is carried out by using mean shift algorithm.

Input: Video \( V \), contains \( N \) frames
Output: Object detection and tracking
Begin
Read Video \( V \);
Step 1 For loop of each video frame \( k = 1 \) to \( N \)
1.1 Read video frame \( V_k \) and \( V_{k+1} \)
1.2 Pick a target area (centre) \( y_0 \)
Step 2 Compute a Normalized weighted 2-D Histogram distribution for area to get the target by using equation (8).
Step 3 Loop
3.1 Get next frame.
3.2 Initialize the location of the target in the current frame with \( y_0 \)
3.3 From the current centre compute the candidate target normalized weighted 2-D Histogram by using equation (10)
3.4 Compute Bhattacharya distance between target and candidate by using equation (7)
3.5 Derive the weights \( \{w_i\}_{i=1}^{nh} \) by using equation (12)
3.6 Based on the mean shift vector, derive the new location of the target,
\[
\sum_{i=1}^{m} x_{ig} \left( \left\| \frac{y_{0} - x_{i}}{h} \right\| \right)
\]  

Update \{p_u(y_i)\}_{i=1...m} and evaluate

\[
\delta[\hat{p}(y_i), \hat{q}] = \sum_{u=1}^{m} \sqrt{p_u(y_i)q_u}
\]

3.7 While

\[
\| \hat{y}_i - \hat{y}_o \| < \epsilon \text{ stop,}
\]

Otherwise set \( \hat{y}_0 \leftarrow (\hat{y}_0 + \hat{y}_i) \) and go to step 1.

End loop

Repeat this till end of frames.

End loop

End of algorithm

5. Accuracy Measure

In order to find the good match between target and candidate propose approach uses Bhattacharyya coefficient. Accuracy of match found can be achieved by three different accuracy measure algorithms [17].

a. Discrete Approximate Match Method of Accuracy
b. Pixel difference accuracy match
c. Pixel cross correlation accuracy match Accuracy

5.1 Discrete Approximate Match Method of Accuracy

INPUT: Two matched images

OUTPUT: The accuracy of match.

1) Let ‘A’ be the reference sub-image and ‘B’ be the matched image.
2) For each position (i,j) in A
3) Find whether there exists a similar point in an area of 3 X 3 around the corresponding position (i,j) in B.
4) Find the total number of match positions.
5) Find percentage of accuracy by, dividing the value of total match by total number of pixels.

\[
Acc = \frac{\text{total match}}{N} \times 100
\]

Where N is the number of pixels.

5.2 Pixel Difference Method of Accuracy

INPUT: Two matched images

OUTPUT: The accuracy of match.

STEPS –

1) Let ‘A’ be the reference sub-image and ‘B’ be the matched image.
2) For each position (i,j) in A
3) Find the difference in pixel value between the reference image and the matched image
4) Find the sum of pixel difference in both images.

\[
\text{Sum} = \sum_{a,b} \text{abs} (A - B)
\]

5) Find percentage of accuracy, by subtracting the sum of difference by the total number of possible values.

\[
Acc = \frac{(255N - \text{sum})}{255 N} \times 100
\]

Where N is the number of pixels.

5.3 Pixel Cross Correlation Method of Accuracy

INPUT: Two matched images

OUTPUT: The accuracy of match.

STEPS –

1) Let ‘I1’ be the reference sub-image and ‘I2’ be the matched image.
2) For each position (i,j) in I1 and I2
3) Find the sum of squared difference between the pixel values in reference and matched image.

\[
\sum (I_{1} - I_{2})^2
\]

4) Find the percentage of difference in pixel value by,

\[
PCC = \frac{\sum (I_{1} - I_{2})^2}{\sqrt{\sum I_{1}} \times \sqrt{\sum I_{2}}}
\]

5) The accuracy of registration is found by subtracting the percentage difference by 100

\[
Acc = 100 - PCC
\]

6. Results And Analysis

The proposed approach is implemented on Windows 7 in Microsoft Visual Studio platform using VC++ language for programming. In our experiments we consider the real time videos taken with different human at various locations and car moving video.

Figure 3: Output of mean shift tracking by using Epanechnikov kernel; searching window size 50X50
Figure 4: Output of mean shift tracking by using Gaussian kernel; searching window size 40X30

Figure 5: Output of mean shift by using normal kernel; searching window size 30X30

Figure 6: Output of mean shift on color feature by using Epanechnikov kernel

Figure 7: Output of mean shift on gaussian kernel

Figure 8: Output of mean shift on uniform kernel

Table 1: shows the measures of Epanechnikov kernel

<table>
<thead>
<tr>
<th></th>
<th>Time</th>
<th>Accuracy</th>
<th>PDM</th>
<th>PCCM</th>
<th>DAMM</th>
<th>TH</th>
<th>TW</th>
</tr>
</thead>
<tbody>
<tr>
<td>ball</td>
<td>1.025000</td>
<td>98.850105</td>
<td>99.597506</td>
<td>100.000000</td>
<td>30</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5.835000</td>
<td>100.000000</td>
<td>100.000000</td>
<td>100.000000</td>
<td>50</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4.776000</td>
<td>98.784000</td>
<td>98.564000</td>
<td>98.809000</td>
<td>60</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>car</td>
<td>2.821000</td>
<td>94.544444</td>
<td>96.796211</td>
<td>98.241846</td>
<td>30</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6.259000</td>
<td>95.356181</td>
<td>93.307196</td>
<td>92.156907</td>
<td>40</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.858000</td>
<td>96.848529</td>
<td>97.623466</td>
<td>91.369545</td>
<td>50</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>human</td>
<td>4.969000</td>
<td>98.050101</td>
<td>99.356827</td>
<td>98.849530</td>
<td>50</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5.029000</td>
<td>98.517647</td>
<td>98.683145</td>
<td>98.065456</td>
<td>60</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4.961000</td>
<td>93.987468</td>
<td>98.368587</td>
<td>99.676500</td>
<td>60</td>
<td>50</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: shows the measures of Gaussian kernel

<table>
<thead>
<tr>
<th></th>
<th>Time</th>
<th>Accuracy</th>
<th>PDM</th>
<th>PCCM</th>
<th>DAMM</th>
<th>TH</th>
<th>TW</th>
</tr>
</thead>
<tbody>
<tr>
<td>ball</td>
<td>8.705000</td>
<td>98.850105</td>
<td>99.597506</td>
<td>97.346078</td>
<td>50</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.385000</td>
<td>99.907098</td>
<td>99.999078</td>
<td>98.238678</td>
<td>50</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5.531000</td>
<td>96.796011</td>
<td>97.564000</td>
<td>99.890000</td>
<td>60</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>car</td>
<td>7.962000</td>
<td>95.844444</td>
<td>97.796211</td>
<td>98.241684</td>
<td>30</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9.940000</td>
<td>94.198618</td>
<td>94.307196</td>
<td>97.156907</td>
<td>30</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.938000</td>
<td>95.866297</td>
<td>97.623466</td>
<td>99.369545</td>
<td>60</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>human</td>
<td>6.579690</td>
<td>98.050401</td>
<td>99.356827</td>
<td>98.649530</td>
<td>50</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6.559000</td>
<td>99.517647</td>
<td>98.848529</td>
<td>99.676500</td>
<td>60</td>
<td>50</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Shows the measures of Uniform Kernel

<table>
<thead>
<tr>
<th></th>
<th>Time</th>
<th>Accuracy</th>
<th>PDM</th>
<th>PCCM</th>
<th>DAMM</th>
<th>TH</th>
<th>TW</th>
</tr>
</thead>
<tbody>
<tr>
<td>ball</td>
<td>2.650000</td>
<td>98.850105</td>
<td>99.597506</td>
<td>97.7745000</td>
<td>30</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.850000</td>
<td>98.850105</td>
<td>99.597506</td>
<td>99.123400</td>
<td>40</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4.120000</td>
<td>98.748501</td>
<td>98.840500</td>
<td>99.809000</td>
<td>50</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td>car</td>
<td>2.019000</td>
<td>94.544444</td>
<td>94.546782</td>
<td>94.241684</td>
<td>30</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6.259000</td>
<td>95.263618</td>
<td>93.307196</td>
<td>92.156907</td>
<td>40</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5.858000</td>
<td>96.848529</td>
<td>97.623466</td>
<td>91.369545</td>
<td>50</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>human</td>
<td>2.571000</td>
<td>98.050401</td>
<td>98.356827</td>
<td>95.542310</td>
<td>30</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4.242000</td>
<td>98.17647</td>
<td>97.88415</td>
<td>97.065456</td>
<td>50</td>
<td>50</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5.460000</td>
<td>98.885466</td>
<td>98.567857</td>
<td>98.358760</td>
<td>60</td>
<td>60</td>
<td></td>
</tr>
</tbody>
</table>

It is observed from table 1 that the Epanechnikov kernel takes less time to track. From table 2 we observe that Gaussian kernel takes more time to track when compared to
Uniform and Epanechnikov kernel. From table 3 it is observed that it takes less time to track when compared with Gaussian kernel. It is observed that from all the tables Epanechnikov kernel gives maximum accuracy when compared to other kernels. Finally from all the tables we can observe that the Epanechnikov kernel will give the best results.

7 Conclusion and Future Work

The proposed approach efficiently tracks the object or human. Mean shift algorithm for tracking works fine with different size of searching window. This method includes color and grey level intensity values. Proposed algorithm has three different kernels which show satisfactory results. Epanechnikov kernel will gives best results. The edge feature is used get the more appropriate result to the proposed method. There are many types of kernels like triangular, tricube, logistic can be used for distribution.

8 Acknowledgements

The authors acknowledge Prof N R Shetty, Director, Nitte Meenakshi Institute of Technology and Dr H C Nagaraj, Principal, Nitte Meenakshi Institute of Technology for providing the support and infrastructure to carry out the research.

References

[1] “Image registration and target tracking” Jharna Majumdhar,Y Dilip ADE
[3] “Mean shift: Robust approach towards feature space analysis” Dorin comaniciu, Peter meer, IEE TRANSACTION Patern matching and machine intelligence, may 2002
[5] “Kernel-Based Object Tracking” Dorin Comaniciu, Senior Member, IEEE, Visvanathan Ramesh, Member, IEEE, and Peter Meer, IEEE transactions on pattern analysis and machine intelligence, vol. 25, no. 5, may 2003
[14] “Robust Scale-adaptive Mean-Shift for Tracking”, Tomas Vojir1, Jana Noskova2, and Jiri Matas1, SCIA 2012
[15] “Constructing re-substitution entropy estimator with discontinuous kernels”, Liyuan Xu ; Su-Juan Wang ; Yu-Lin Hg, IEEE July 2013