according to their appearance likelihood ratio between the target model and the candidate distribution.[1] To locate the new position of object these pixel weights are used. Using these pixel weights density gradient are calculated in the image coordinates. In this approach mean shift iterations are used for tracking.

1.2. Mean *Shift Tracker*

Mean shift is a non-parameter density estimation method. By iterative searching it finds the maximum similar distribution pattern with sample pattern. Mean shift is applied for target tracking by calculating similarity between features of target model and candidate model.[2] Mean shift algorithm has two parts. In one part it prepares a target appearance descriptor and in second part it tracks the target object using mean shift tracking. In classic mean shift similarity between target color histogram and candidate color histogram is calculated. In this at first frame target color histogram is prepared. After this the candidate area which has maximum similarity with target area is found. To calculate maximum similarity mean shift is used. Steps of mean shift tracking are: extract the candidate region which is most similar to target region, extract color histogram of candidate region calculate similarity measure function, calculate shift vector. Using non-parameter density estimation shift vector maximize the similarity between target histogram and candidate histogram. Mean shift algorithm is iteratively converge to the maximum similar candidate area with target.

Classic mean shift has some advantages as it gives good real time performance, good search efficiency; it is insensitive to non-rigid deformation, partial occlusion and overlap. It uses color histogram for feature representation which is invariant to rotation. Color histogram is also robust to partial occlusion. Classic mean shift also has advantage of color histogram. It has some drawbacks due to fixed kernel bandwidth. When the target scale changes tracking of target is failed. When there is color aberration due to light changing and similar color background then there is problem of target tracking. Some improved methods are introduced in reference 4, 5, 6. There is another drawback of classic mean shift is due to appearance description. It uses color histogram as appearance descriptor; color features cannot provide enough information about the target. To fix this problem for appearance description with color histogram another feature descriptor is used. In an improved version of mean shift texture features are also used with color histogram for appearance description. Color histogram and feature descriptor are complementary to each other means when one is fail to track some features of the target then other one can track it.

2. Mean Shift Algorithm

Mean shift algorithm is performed mainly in two parts. In one part target appearance model is prepared and in another part tracking is take place.

2.1. *Target Appearance*

Firstly target appearance model is prepared. This model is further used for target tracking by estimating maximum similarity between target model and candidate model.

Target representation in classical mean shift is done using color histogram, but in advanced mean shift target is represented using texture features and color histogram.

2.1.1. *Color Feature Descriptor*

Color features of target are described using color histogram. In color histogram each sub-space of RGB spaces is divided into k-intervals known as bins. Each interval consists of feature space. On the basis of data of pixels in target area probability of each bin is calculated. This probability is integrated into color histogram. After integration each pixel in target region are given a weight value. Inner pixels are more distinguishable than outer because of interference created by background. So inner pixels have given higher weight than outers.

Total number bins in feature space are $m_c=k^3$.

The probability density of color feature space values u=1…….m_c is calculated as follows:

$$\hat{q}_{c_u} = C \sum_{i=1}^{n} k\left(\left\|\frac{x_0-x_i}{h}\right\|^2\right)\delta[c(x_i)-u] \qquad (1)$$

“Eq. (1)” is kernel density estimation expression. x_0 is the center of target area.

x_i =1………n are n pixels of the region.

k() is the monotone decreasing profile function. $\delta(x)$ is the delta function.

Role of $\delta[c(x_i) - u]$ is to find whether color value of x_i belongs to u-th bin or not. It returns value 1 if it belongs to u^{th} bin otherwise 0.

C is the normalization constant.

It is calculated as

C= $\frac{1}{\sum_{i=1}^{n} k(\left\|\frac{x_0 - x_i}{h}\right\|^2)}$ to ensure $\sum_{u=1}^{m} q_{c_u} = 1$.

After calculating target appearance model, in later frames candidate appearance model is calculated.

$$\hat{p}_{c_u}(y) = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right) \delta[c(x_i) - u] \qquad (2)$$

Where y is center of candidate region.

$C_h = \frac{1}{\sum_{i=1}^{n} k(\|\frac{y - x_i}{h}\|^2)}$ is normalization constant.

2.1.2 *Texture Feature Description*

Texture feature description gives texture features of the target appearance and candidate appearance which is used in calculating similarity between target model and candidate model. These features are used with color histogram.

In this paper two approaches for texture feature description are compared:

(i) Discrete Wavelet Transform

Discrete wavelet transform is used to extract texture feature of image. Using extracted wavelet coefficient histogram is calculated to adapt the mean shift framework.

It is an effective way in multi-resolution image processing. In this method first the image is decomposed into sub-image with different multi-resolution space and independent frequency band. Further processing is done on sub-image coefficient. In four sub-image of an image three are high-frequency sub-image and one is low frequency sub-image. For further processing low-frequency sub-image is used to produce four sub-images.

Statistical characteristic of low-frequency sub-image are similar to original image. High frequency sub-images represent the edges and texture of image.

Discrete wavelet coefficient of an image is calculated using following equation:

$$W_{sd}(x,y) = \frac{1}{\sqrt{N}} \sum_{y \in ROI} I(y) V_{sd}(y) \qquad (3)$$

Where V_{sd} the wavelet function, s is the scale, d is the direction number.

First obtain s*d sub-spaces of wavelet coefficient. After this divide each sub-space into j-intervals and calculate the probability of $m_t = s * d$ bins.

After this probability density of texture feature space values u=1…m_t is estimated as follows:

$$\hat{q}_{t_u} = C \sum_{i=1}^{n} k\left(\left\|\frac{x_0 - x_i}{h}\right\|^2\right) \delta[t(x_i) - u] \qquad (4)$$

Candidate region's probability density of texture feature space values u=1…m_t is estimated as follows:

$$\hat{p}_{t_u}(y) = C_h \sum_{i=1}^{n_h} k\left(\left\|\frac{y - x_i}{h}\right\|^2\right) \delta[t(x_i) - u] \qquad (5)$$

(ii) Local Binary Pattern

The LBP operator labels the pixel in an image by thresholding its neighbor-hood with the center value and considering the result as a binary number (binary pattern). The general version of the LBP operator is defined as follows:

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \qquad (6)$$

g_c corresponds to the gray value of the center pixel, (x_c, y_c) is center pixel of a local neighborhood.

g_p is the gray values of *P* equally spaced pixels on a circle with radius *R*.

Function s(x) is defined as:

$$S(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$$

The gray-scale and rotation invariant LBP texture model is obtained by

$$LBP_{P,R}^{riu2} = \begin{cases} \sum_{p=0}^{P-1} s(g_p \quad g_c) & if\ U(LBP_{P,R}) \leq 2 \\ P + 1 & otherwise \end{cases} \qquad (7)$$

Where

$$U(LBP_{P,R}) = |s(g_{P-1} - g_c) - s(g_0 - g_c)| + \sum_{p=1}^{P-1} |s(g_p - g_c) - s(g_{p-1} - g_c)|$$

"*riu2*" means that the rotation invariant "uniform" patterns have a U value of at most 2. By definition, the P +1 "uniform" binary pattern occur in a circularly symmetric neighbor set of P pixels. Above equation assigns a unique label to each of them corresponding to the number of "1" bits in the pattern (0 to P), while the "non uniform" patterns are grouped under the "miscellaneous" label (P + 1).

In order to make LBP more robust against subtle changes in pixel values, the thresholding strategy in the LBP operator is modified by replacing the term $s(g_p - g_c)$ with $s(g_p - g_c + a)$. The greater the value of $|a|$ is, the higher fluctuations in pixel values are allowed without affecting much the thresholding result. The LBP feature of each point in the image region, whose value is between 0 and 9 is calculated. Thus an appearance model combining the color and texture is constructed and it consists of color channel and LBP texture pattern. The $LBP^{riu}_{8,1}$ model has nine uniform texture patterns. Each of the $LBP^{riu}_{8,1}$ uniform patterns is regarded as a micro-texton. The local primitives detected by the $LBP^{riu}_{8,1}$ model include spots, flat areas, edges, line ends and corners, etc. In target representation, the micro-textons such as edges, line ends and corners, by name of "major uniform patterns", represent the main features of target, while spots and flat areas, called "minor uniform patterns", are minor textures. The main uniform patterns of the target are extracted by the following equation:

$$LBP^{riu2}_{8,1} - \begin{cases} \sum_{p-0}^{7} s(g_p - g_c + a) & U(LBP_{8,1}) \leq 2 \text{ and } \sum_{p=0}^{7} s(g_p - g_c + a) \in \{2,3,4,5,6\} \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

In $LBP^{riu}_{8,1}$, the labels corresponding to minor uniform patterns are 0, 1, 7 and 8 respectively, and the label of non uniform patterns is 9. The labels corresponding to main uniform patterns are 2–6, which have five patterns. This equation groups the minor uniform patterns as non uniform patterns. Generally, the main LBP features of a target are more important than its minor features to represent the target. First this equation is used to form a mask and then the color and LBP features within this mask are used to model the target appearance model.

2.2 Target *Tracking*

With the descriptions of target and candidate, the similarity between them can be calculated using similarity measure function. The tracking process is achieved by the search in the current frame to maximize the similarity function to obtain the location of target. The color texture integrated similarity is defined as:

$$\hat{\rho}_{com}(y) \equiv \sum_{u=1}^{m_c} \sqrt{\hat{p}_{c_u}(y)\hat{q}_{c_u}} \sum_{u=1}^{m_t} \sqrt{\hat{p}_{t_u}(y)\hat{q}_{t_u}} \quad (9)$$

To maximize $\hat{\rho}_{com}$, the target central position in last frame y_0 is taken as initial candidate center in current frame, then searching the best match of the target from this initial center. Then based on Taylor series expansion of the integrated formula $\hat{\rho}_{com}$ can be approximated as:

$$\hat{\rho}_{com} = \frac{C_h}{2} \sum_{i=1}^{n_h} w_i k(\| \frac{y-x_i}{h} \|^2) \quad (10)$$

Where

$$w_i = \sum_{u=1}^{m_t} \sqrt{\hat{p}_{t_u}(y)\hat{q}_{t_u}} * \sum_{u=1}^{m_c} \sqrt{\frac{\hat{q}_{c_u}}{\hat{p}_{c_u}(y_0)}} \delta[c(x_i) u] + \sum_{u=1}^{m_c} \sqrt{\hat{p}_{c_u}(y)\hat{q}_{c_u}} * \sum_{u=1}^{m_t} \sqrt{\frac{\hat{q}_{t_u}}{\hat{p}_{t_u}(y_0)}} \delta[t(x_i) - u]$$

Using mean shift algorithm, the candidate center y_1 re-positioned in each iteration is calculated as follow:

$$y_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g(\| \frac{y_0 - x_i}{h} \|^2)}{\sum_{i=1}^{n_h} w_i g(\| \frac{y_0 - x_i}{h} \|^2)} \quad (11)$$

3. Comparison of LBP and DWT

Anyone of the LBP and DWT can be used for texture feature extraction in video tracking. Features extracted using one of these methods is are used with color histogram for target tracking. Texture features are integrated with color histogram; integrated features are used for feature description of target region and candidate region.

In this paper LBP and DWT are compared. This paper concludes which one is better for texture feature extraction in video tracking.

3.1. *Outputs showing the difference between two methods*

Figure 1 shows output using LBP and figure 2 shows output using DWT of same image.

Fig 1 Output of LBP method

Fig 2 Output of DWT method

As it can be seen from the above figures DWT gives better output in comparison of LBP.

Computing time taken by LBP is 0.102669 seconds and time taken by DWT is 0.225109 seconds.

Table 3.1 Comparison of LBP and DWT

Features	LBP	DWT
Output	goo d	Better than LBP
Implementation	Difficult in comparison	Easy
Computing Time	Less time consuming	More time consuming

As it can be seen from the characteristic comparison of the two methods DWT gives better output and it is easy to implement in comparison of LBP. Whereas LBP takes less computing time in comparison of DWT. Computing time difference is very less so this comparison concludes that DWT gives better texture extraction in comparison of LBP.

4. Conclusion

Video tracking using features of target, need more information other than provided by color histogram for better output. Color histogram has some limitation such as similar background, to overcome these limitations texture features are too used for the feature description. This paper compares two approaches of texture feature extraction one is LBP (Local Binary Pattern) and other is DWT (Discrete Wavelet Transform). It concludes that DWT is better than LBP as it gives better output.

5. References

1. Comaniciu, D., Ramesh, V., Meer, P.,*Real-time tracking of nonrigid objects using mean shift* (IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 142–149, 2000)

2. Fukunaga, K., Hostetler, L.D., *The estimation of the gradient of a density function, with applications in pattern recognition* (IEEE IT 21(1), 32–40 ,1975)

3. Comaniciu, D., Ramesh, V., Meer, P., *Kernel-Based Target Tracking* (IEEE Trans.Pattern Analysis and Machine INtelligence,vol.25, no.5, pp.564-577,2003)

4. T.L.Liu, and H.T. Chen, *Real-time tracking using trust-region methods* (IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, No. 3,pp. 397-402,2004)

5. R.T. Collins, *Mean-shift Blob Tracking through Scale Space*, (IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2,pp. 234-240,2003)

6. Z. Zivkovic and B. Krose, *An EM-like algorithm for color-histogram-based target tracking* (IEEE Conference on Computer Vision and Pattern Recognition, pp.789-803,2004)

7. Xiang-Zhang, Yuan-Ming Dai, Zhang-Wei Chen, Huai-Xiang Zhang, *An Improved Mean Shift Tracking Algorithm Based On Color And Texture Feature* (Proceedings of the 2010 International Conference on Wavelet Analysis and Pattern Recognition, Qingdao, 11-14 July 2010)

8. Jifeng Ning, Lei Zhang , David Zhang, Chengke Wu, *Robust Object Tracking Using Joint Color-Texture Histogram* (International Journal of Pattern Recognition and Artificial Intelligence Vol. 23, No. 7 (2009) 1245–1263).

9. Pratibha Sharma, Manoj Diwakar, Niranjan Lal, “Edge Detection using Moore Neighborhood”, International Journal Of Computer Applications, Volume 61– No.3, January 2013, Pages 26-30.

10. Pratibha Sharma, Manoj Diwakar, Sangam Choudhary, “Application of Edge Detection in Brain Tumor Detection”, International Journal Of Computer Applications, Volume 58– No.16, November 2012, Pages 21-25.