**Final Report**

**On**

**Motion Estimation Via cluster matching**

**By**

**Pravin Reddy Sereddy**

**B-number: B00714266**

**Table of Content**

1. Abstract
2. Introduction
3. Methodology
4. Results
5. Conclusion and Discussion
6. References

**Abstract:**

The Goal of this project is to find the motion estimation in the images using the k means clustering algorithm by using only the few features like position and the intensity without using the any object detection or brightness or any machine learning algorithm.

The motion estimation in this model is done in two steps that is

In the first step we do the clustering of the pixels of each frame independently with the features like position and intensity. This process is done using the k means algorithm for the clustering.

In the second step we match each cluster centres fron one frame to another using the features like the position, intensity, momentum, average greyscale difference between the clusters.

After the cluster matching, we calculate the distance between matching cluster centres from one frame to another frame and if the values are greater than the threshold value then motion vectors are drawn on the image frame to determine the motion estimation.

**Introduction:**

Motion Estimation is the process of defining the motion vectors that describe the transformation from one two-dimensional image frame to another two-dimensional adjacent frame in a video sequence.

Motion estimation is very useful technique in computer vision in this modern era.

It is used in many applications like autonomous navigation for the determining the direction of the vehicle is moving and it is also used in the satellite imaginary analysis to predict the weather for determining the storm cloud movement and is also used in the medical imaging.

One of the main applications is used is for the video encoding to remove the temporal redundancy using the motion estimation and compensation to compress the video.

In the previous years many researches have done the research work on the motion estimation based on the object detection and the illumination conditions, but this model is based on the simple pixel clustering using the features like position and the intensity on the pixels in the image sequence frames. In this model you need to know little amount of knowledge is required to estimate the motion.

This method is applicable for all the images irrespective of the image pixels, object shape, brightness and contrast of the object. That’s why this method is more robust than any other method.

**Methodology:**

Motion estimation via cluster matching is the two-step process

1. In the first step we have to cluster the pixels in each image frames independently using the modified k means clustering algorithm by using the features like position of the pixel and the intensity value of the pixel
2. In the second step we have to match each cluster from the cluster set from the first image frame to each cluster in clusters set of the second image frame

After matching the cluster then we find the displacement of the cluster centres and draw the motion vectors from these displacement values of the cluster centres.

***Clustering using the k means algorithm:***

In this algorithm we first take all the pixels from the image frame and we form a 3x1 matrix with the x coordinate, y coordinates the intensity of the pixel and we take all these 3x1 matrix for each pixel [1].



Fig 1: pixel in matrix [1]

Initially we assign the k value and we divide the image into these k square regions and considering each region as a cluster initially and from these clusters we find the centroid of the cluster by the mean value of the x coordinates in the cluster, y coordinates in the cluster and the intensity values in each cluster. For each cluster we take the mean x, y coordinate and intensity values and form the 3x1 matrix [1].

Fig 2: cluster means matrix [1]

Now find the distance for each pixel to each cluster by using the weighted square Euclidean distance to assign each pixel to the cluster with the smallest distance value. By doing this we find the new cluster centres whenever a new pixel set form in the cluster. This process is iterated until the all the cluster centres are not moved.

The formula for the weighted square Euclidean distance formula [1].



Fig 3: weighted square Euclidean distance formula [1]

Here i is the number of pixels and j is the number of clusters and the Wj [1] is

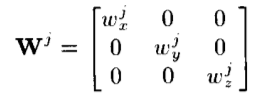


Fig 4: weight matrix [1]

The weights Wjx, Wjy, Wjz are used to reduce the intra set distance between the pixels and we find the weights Wjx, Wjy, Wjz by using this formula [1].

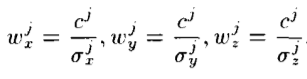




Fig 5: weights formula [1]

We find the Unbiased variance estimate of the pixels in a cluster in x direction by using the

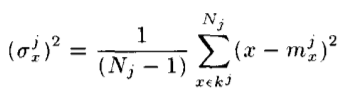


Fig 6: variance formula [1]

Similarly, we find variances for the y and z values and the purpose of the weight matrix is to balance the emphasis of the position to the intensity in the cluster.

After the iteration of this process is stopped now, we start the cluster matching to match the clusters

***Clustering matching:***

After calculating the clusters we do the cluster matching analysis in order to match the clusters from first image frame to second image frame

Let’s consider the cluster set for the first image frame at time t



And let’s consider the cluster set for the first image frame at time t + dt



In the cluster matching to find the accurate cluster matching. We consider two new features to the matrixes. First feature we add is average grey scale value difference in the cluster and the second feature is the sum of squared central momentum in x, y direction.

Let bj be the vector of matching features for cluster kl and, cl denote the vector of matching features for kl and the mj is the the mean x, y coordinate and intensity values of the cluster in the form of 3x1 matrix.

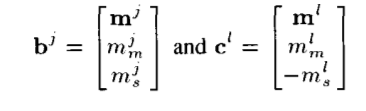


Fig 7: cluster matching feature matrix [1]

To find the average grey scale value difference in the cluster is

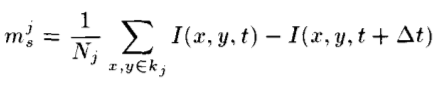


Fig 8: average greyscale value difference formula [1]

To find sum of the squared central momentum of the cluster in x, y direction

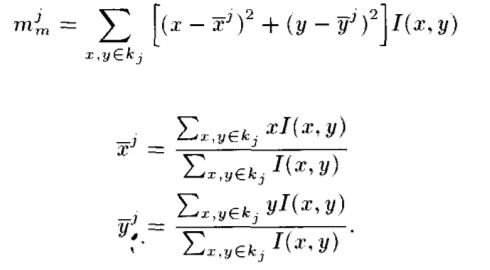


Fig 9: sum of the squared central momentum of the cluster formula [1]

***How does these features help in cluster matching?***

* For each cluster with different intensities, but the same shape and equal displacements, the mean intensity values (mz) influence the matching clusters
* If the moving clusters have the same intensities and equal displacements but different shapes, the values of the squared central moments (mm) which gives the shape information helps in matching
* Finally, when clusters have the same shape and intensity, but one is stationary, while the other moves, the difference averages (ms) help determine the correct match

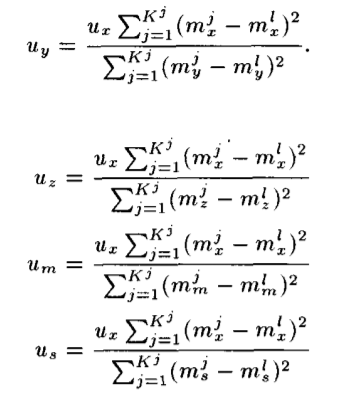
***How to find the cluster matching value?***

We find the cluster matching values by the weighted squared Euclidean distance measure



For every cluster ‘i’ in the first frame is matched with every cluster ‘j’ in the second frame and we match the cluster with least distance Djl.

**U** is the diagonal weight matrix with diagonal values of ux, uy, uz, um, us and rest all values are zeros



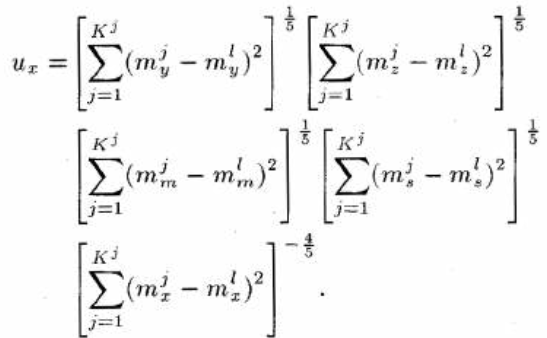
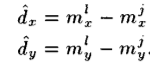


Fig 9: weight matrix values formula [1]

***How to show the motion vectors?***

After finding the cluster matches, we take the displacement between the matching cluster centres and draw the vectors from the data we got.



**Results:**

**Example 1**

Input image frames:



Fig 10: frame1



Fig 11: frame2

Cluster center results:

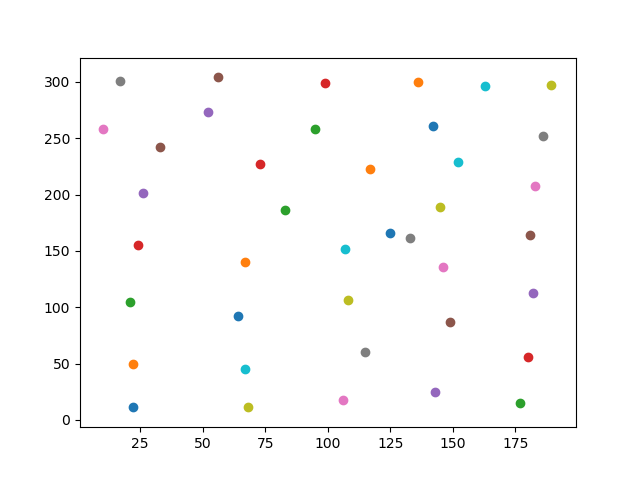


Fig 12: frame1 cluster centres after clustering

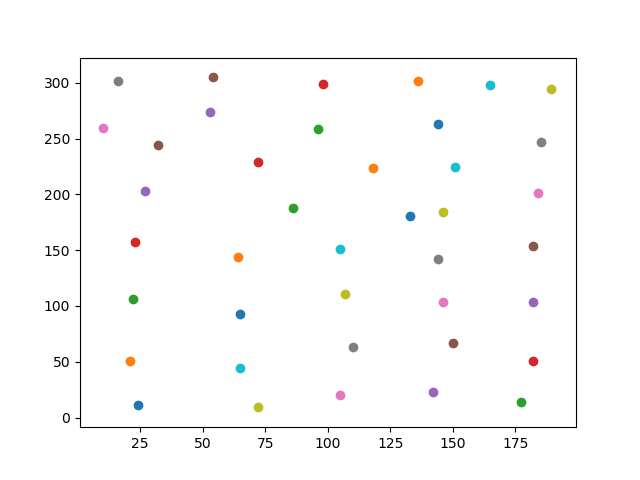


Fig 13: frame2 cluster centres after clustering

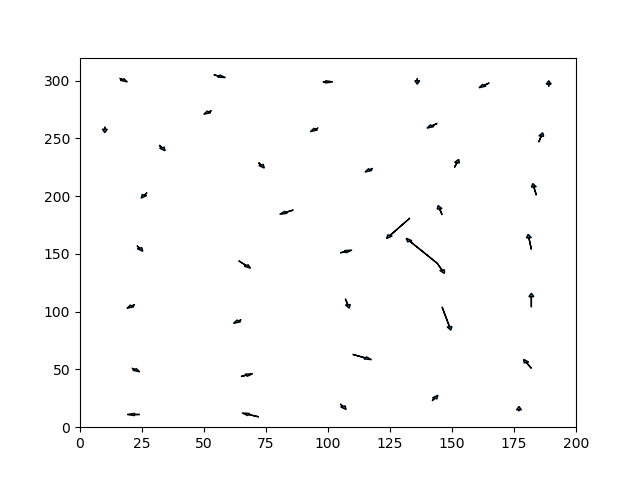


Fig 14: Motion vectors image



Fig 15: frame1 cluster centres after clustering

As we can see from the two frames the ball goes up and hand goes down and motion vectors shows the output

**Example 2**

Input image frames:



Fig 10: frame1



Fig 11: frame2

Cluster center results:

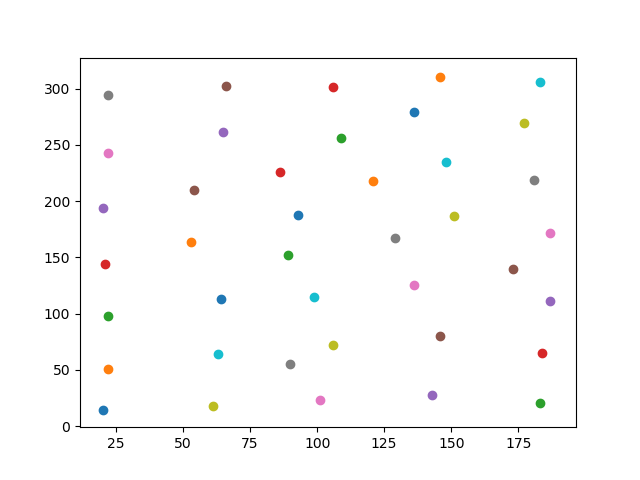


Fig 12: frame1 cluster centres after clustering

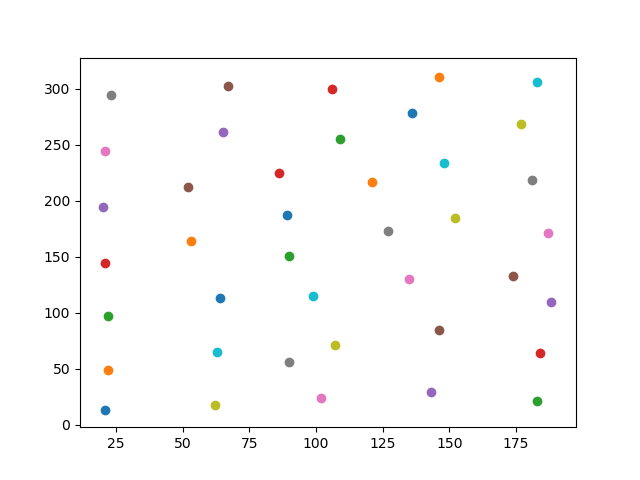


Fig 13: frame2 cluster centres after clustering

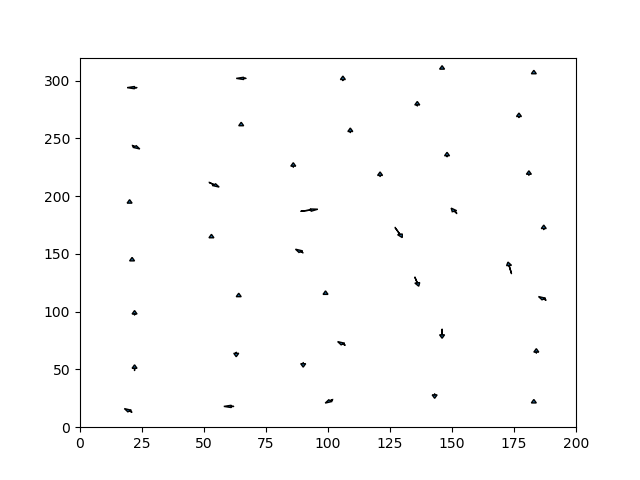


Fig 14: Motion vectors image



Fig 15: frame1 cluster centres after clustering

Due to very little moment in the example 2 there is very little moment of vectors in this example

**Conclusion and Discussion:**

In this project I have developed a cluster matching motion estimation in python by analysing the image using the features like position, intensity, momentum, average greyscale value difference.

This approach is very different approach from the current techniques and this process is done in two steps first is clustering with position and intensity using the k means and in the second step we take the position, intensity, momentum, average greyscale value difference for the cluster matching and drawing the vectors from the matched clusters

In future we can make this algorithm flexibly embedded into the autonomous navigation for detecting the direction of the object and in the can be embedded in the medical and the weather satellite systems for the motion estimation

**References:**

**[1]** Dane P. Kottke, Ying Sun, “Motion Estimation Via Cluster Matching” IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL. 16, NO. 11, November 1994

**[2]** J. K. Agganval and N. Nandhakumar, “On the computation of motion from sequences of images-A review,” Proc. IEEE, vol. 76, pp. 917-935, 1988

J. Tou and R. C. Gonzalez, Pattern Recognition Principles. Reading, MA: Addison-Wesley, 1974

**[3]** https://en.wikipedia.org/wiki/Motion\_estimation