Video Scene Segmentation with Histogram Distances and Edge Change Ratio

Jack Robiou
New York University
New York, USA
jir285@nyu.edu

Marcus Robiou
New York University
New York, USA
mp3741@nyu.edu

Su Lei Win
New York University
New York, USA
sw5205@nyu.edu

Wilson Li
New York University
New York, USA
tl2894@nyu.edu

I. Introduction

Videos have an inherent hierarchical structure. A video consists of single or multiple stories. Once a video has split into scenes, each scene is composed of one or more consecutive shots from multiple camera angles in the same location or at the same time. We will have a new scene when there is a change in time or location. Each shot is recorded from a fixed camera angle and it is made up of many frames. Each frame represents a single image. The hierarchical structure of a video is shown in Figure 1.

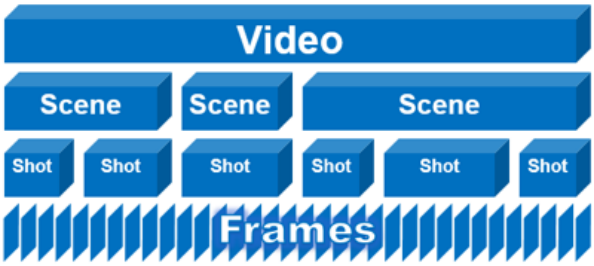


Fig. 1. Hierarchical structure of a video

Among them, the scene level is the best to use for browsing and retrieving the video because it provides a complete understanding of video content. However, segmenting a scene is a challenging problem due to different shooting and editing styles in the media and entertainment industry. Therefore, we would like to test different scene segmentation methods and then asses and compare each method's strengths and weaknesses. Scene segmentation is a method to detect the point of transition between scenes and split them accordingly. There are different scene segmentation methods that can be used for this purpose. We decided to use histogram comparisons and edge change ratio as the two main approaches for this project. The rest of the paper will be organized as follows: Section II approach, Section III and IV describe the Histogram approach and the Edge Change Ratio approach with their respective experiment, result and analysis.

II. Approach

A. Dataset

To compare and evaluate both of our methods, we use an Open Video Scene Detection (OSVD) open dataset from IBM, which includes 21 movie video clips with various length and genre content. Each video is approximately 10 - 20 minutes long with 10k to 100k frames.

B. Pre-processing

Each clip used during experimentation was processed by performing the following steps:

- Extract and store each individual frame as an image.
- Downscale each frame by a fixed percent (50% in most cases) while preserving their original aspect ratio. This step was necessary for about 80% of the clips used, as most videos were too large to perform the other pre-processing steps on each of their individual frames without causing memory issues.
- Convert each extracted frame to grayscale, which allows for easier comparisons of their respective histograms as there would only be one intensity channel to consider.
- After this step, there is a different processing step for each method.
- For the edge change ratio method, we detect the edge of the image from the gray scale frames and count the number of edge pixels. After that, we dilate and invert binary images into corrosion edges. In the end, we count the number of entering edge pixels,
- For the histogram comparison approach, we scale and normalize the intensity values of each frame to a range between 0 and 255 to ensure each image histogram has an equal number of bins in the same range of values.
- Reduce noise in each frame by applying a Gaussian filter while using a value of 5 for the sigma parameter.

The following subsection provides a more detailed description of grayscale conversion and noise reduction:

- Grayscale Image Processing:
  Grayscale is a range of monochromatic shades from black to white, so grayscale only contains shades of gray and no color. In the real world, we call it a black and white image. The value of each pixel in a grayscale image carries only intensity information. The image frames we extracted from the videos are in Red, Green, Blue (RGB) colors, so each pixel has three separate luminance values. Those values will be combined into a single value when converting to a grayscale image. In other words,

grayscale image processing removes all color information and leaves only the luminance of each pixel.

- Gaussian Filter:
  A Gaussian blur (also known as Gaussian smoothing) is a linear filter. It is used to blur images and reduce the noise of the image. Gaussian smoothing can be computed by using standard convolution methods. For the larger standard deviation, we have to convolve an image several times. Before we apply any of our scene segmentation methods, we used a Gaussian filter to reduce the level of noise in the image.

C. Histogram Distances

This approach to scene segmentation relies on, first, calculating the intensity histogram of each individual frame of a given movie clip, and, next, determining similarity between frames by calculating the distance between their respective histograms in sequential order. Next, a predetermined threshold is used to identify whether each subsequent frame belongs to the same scene as that of the previous frame, or if it belongs to an entirely new scene. In this way, scenes are ultimately segmented based on differences in their histograms, which correlates to differences in intensity values.

There are multiple different distance functions which may be used to determine the similarity between histograms, and for this project four main metrics were used:

1) **Correlation**:
   This distance metric is calculated by multiplying the covariance of two variables and dividing it by the product of their respective standard deviations. This means it has the same properties as the covariance, and thus is an effective way to determine the linear relationship between two variables (or in our case two histograms).

$$d(H_1, H_2) = \frac{\sum_I (H_1(I) - \bar{H}_1)(H_2(I) - \bar{H}_2)}{\sqrt{\sum_I (H_1(I) - \bar{H}_1)^2 \sum_I (H_2(I) - \bar{H}_2)^2}}$$

   As with the covariance, the mean of each histogram must be calculated for the correlation computation:

$$\bar{H}_k = \frac{1}{N} \sum H_k(J)$$

2) **Chi-Square**:
   The Chi-Square distance is based on Pearson's chi-squared test, and is a statistical test which can be used to determine the difference between expected frequencies and observed frequencies across two histograms. Greater similarity is assigned to smaller differences, i.e., pairs of histograms where differences in frequencies are more often expected than not. It should be noted that this particular formula is asymmetric.

$$d(H_1, H_2) = \sum_I \frac{(H_1(I), H_2(I))^2}{H_1(I)}$$

3) **Intersection**:
   This simple distance metric finds where two histograms meet by taking the sum of the minimum values between each pair of bins. Histograms with no intensity values in common would yield an intersection distance value of 0.

$$d(H_1, H_2) = \sum_I min((H_1(I), H_2(I))$$

4) **Bhattacharyya**:
   The Bhattacharyya distance makes use of the Bhattacharyya coefficient to obtain the similarity between two histograms. It works by making use of the Bhattacharyya coefficient, which can measure the amount of overlap between two statistical samples, and can also be applied to histograms as they represent a probability distribution of intensity values in an image.

$$D(x, y) = 1 - \sqrt{\sum_{i=1}^{n} \frac{\sqrt{x_i y_i}}{\sqrt{\sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}}}$$

(Note that there are multiple formulas for calculating some of these distances)

Each distance function requires a different threshold to be effective, and two of them (correlation and intersection) determine more similarity the greater the distance, as opposed to the more usual way of smaller distances relating to increased similarity.

Illustration

Predetermined scene annotations were used as reference to achieve the desired segmentation, however, during testing it was found that these annotations were in many cases defined arbitrarily, and thus are not a reliable way to measure the accuracy of our results.

After pre-processing, segmentation was run on all frames using each distance metric, while the optimal threshold was found by attempting to create an output size which most closely matched that of the annotations.

A test run is conducted on the following video.

- Clip Name: Big Buck Bunny
- Frame Count: 14314

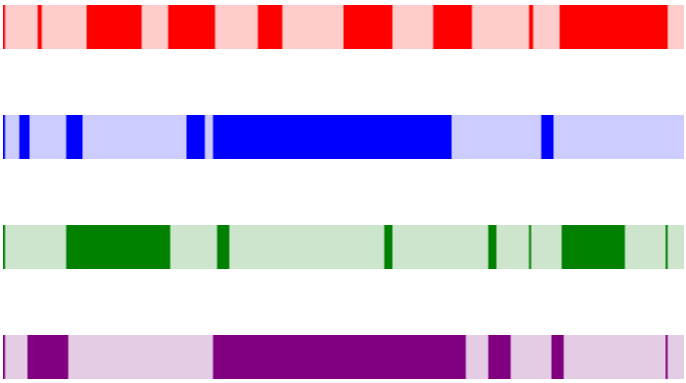
Each entry in the above chart corresponds to a separate scene in the clip, while the numbers of the start and end frames represent each scene. As can be observed, the results are quite different from the given scene annotations, so, for a better idea of whether the scene segmentation is properly separating frames based on their intensity values, here are some visualizations of the start and end frames of segmented scenes.

Given Annotation Labels:

Correlation Method:

Chi Square Method:

Annotations	Correlation	Chi Square	Intersection	Bhattacharyya
[50, 749]	[50, 377]	[50, 1345]	[50, 552]	[50, 552]
[811, 1812]	[378, 552]	[1346, 3512]	[553, 1345]	[553, 1345]
[1813, 2900]	[553, 1345]	[3513, 4543]	[1346, 4462]	[1346, 4462]
[2901, 3512]	[1346, 1665]	[4544, 4725]	[4463, 9718]	[4463, 9718]
[3513, 4462]	[1666, 3894]	[4726, 8023]	[9719, 10233]	[9719, 10233]
[4463, 5370]	[3895, 3918]	[8024, 8153]	[10234, 10302]	[10234, 10302]
[5371, 5843]	[3919, 3988]	[8154, 10233]	[10303, 10338]	[10303, 10338]
[5844, 7164]	[3989, 4031]	[10234, 10338]	[10339, 10636]	[10339, 10636]
[7224, 8153]	[4032, 4068]	[10339, 11059]	[10637, 11559]	[10637, 11559]
[8154, 9079]	[4069, 4226]	[11060, 11060]	[11560, 11763]	[11560, 11763]
[9080, 9849]	[4227, 4462]	[11061, 11763]	[11764, 11764]	[11764, 11764]
[9850, 11052]	[4463, 9419]	[11764, 13025]	[11765, 13919]	[11765, 13919]
[11102, 11725]	[9420, 11311]	[13026, 13919]	[13920, 13941]	[13920, 13941]
[11726, 13941]	[11312, 11559]	[13920, 13941]	[13942, 14313]	[13942, 14313]
[13942, 14314]	[11560, 14314]	[13942, 14313]	[14314, 14314]	[14314, 14314]



Intersection Method:
Bhattacharyya Method:

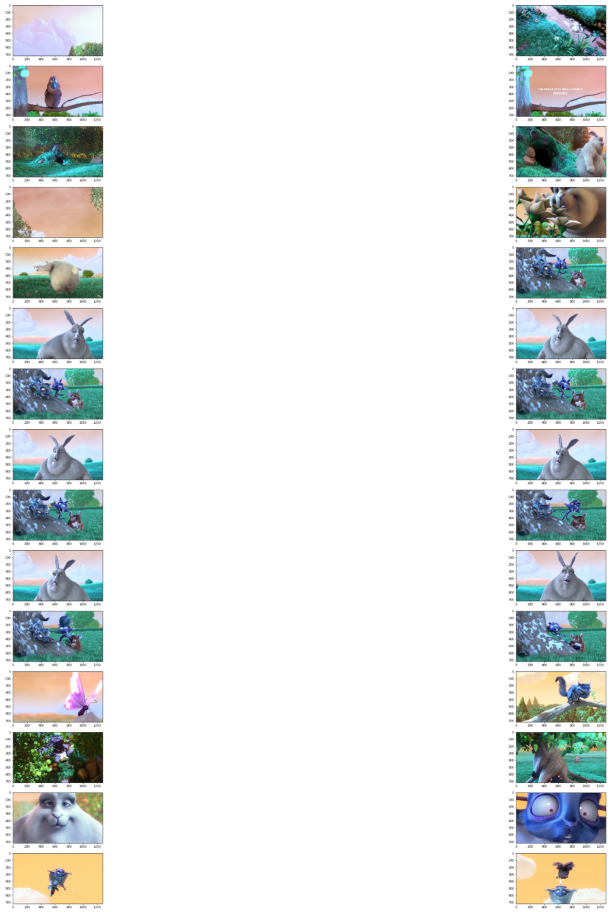


Fig. 2. Start and end frames of each segmented scene via correlation

Qualitative Analysis

As can be seen here, there is quite a bit of variation in appearance from one end of each scene to the start of the next scene, giving us the intuition that our segmentation implementation is indeed separating scenes by how much their intensity histograms vary. Nonetheless, certain segmented "scenes" appear more closely to be shots, which are much more closely related to camera shifts than an actual change in setting.

In these segmented scenes, it is clear that the camera is actually going back and forth between two points of view, which indicates that in reality it all belongs to a single scene. This highlights the challenge of segmenting scenes when compared to segmenting shots. While shots are usually defined as whenever the camera makes an abrupt shift in its point of view (for instance a jump cut), scenes can be a bit more loosely defined.

Although many clips feature scene transitions as a way to go from one scene to another, there plenty of instances where these transitions are not as clearly defined, thus, it is quite possible that any segmentation method which relies purely on intensity values of each frame (as is the case with histogram comparison) may be more suited for segmenting shots rather than scenes, especially given that each and every shot will be distinct in its background and/or characters.

It is also important to note that the distance metrics used for the histogram comparisons do not take into account cross-bin similarity, which is to say they forego bin to bin comparisons within the same frame, only comparing bins from one frame to another. This may also have an effect on the resulting segmentation, as a metric which takes into account cross-bin similarity will assign greater importance to particular bins within each frame, allowing for identification of the specific intensity values which better represent a given frame and thus resulting in more nuanced comparisons.

Overall, it appears that, for proper scene segmentation, the method employed should somehow account for more abstract differences between frames such as semantics, something which may be accomplished via more complex technologies such as convolutional neural networks. Based on the results obtained from segmenting various scenes and clips via histogram comparisons, it appears that a segmentation method which exclusively uses visual identifiers such as intensity values to group frames into scenes yields an output more closely resembling shots instead, which have a clearer and

Fig. 3. Camera going back and forth between two points of view

less arbitrary definition than scenes do.

D. Edge Change Ratio

The idea behind the Edge Change Ratio (ECR) method is that scene shifts will have a hard cut between the edges of the old frame's objects and the new frame's objects. We detect this structural discontinuity through the edge change ratio by calculating the amount of pixels that moved more than a fixed distance away from where it previously was. Between two consecutive frames, pixels are either entering or exiting so we can utilize the value of the largest incoming/outgoing change to detect hard scene shifts in movies. The edge change ratio between the two scenes is calculated by using the formula:

$$ECR_n = max(\frac{X_n^{in}}{\delta_n}, \frac{X_{n-1}^{out}}{\delta_{n-1}})$$

δ_n is the number of edge pixels in frame n which is also just the number of pixels in a gradient image. X_n^{in} and X_{n-1}^{out} are the number of entering and exiting edge pixels in frames n and n-1 respectively. X_n^{in} and X_n^{out} are calculated as followed:

$$X_n^{in} = int(e_n \& d_{n-1})$$

$$X_n^{out} = int(e_{n-1} \& d_n)$$

where the number of entering or exiting pixels in frames n and n-1 are respectively $X\_OR'd$ with the number of pixels in the corrosion image from frames n-1 and n.

Illustration

We tested on the "fires_beneath_water" video clip for this experiment. We extracted each individual video frame and subsequently scaled the image, calculated the image gradient, dilated the image, and finally inverted it to obtain the image corrosion. This can be viewed in Figure 4 below. Based on the above equation, we calculate the ECR using the number of pixels in the gradient and corrosion image which finds the ratio of edge pixels that moved over the total number of pixels in the frame. Plotting the image would depict all of the edges that have moved from their location in the previous frame to the current frame.

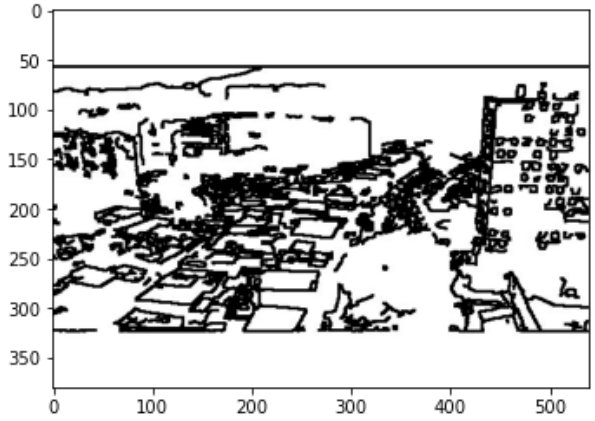


Fig. 4. Edge Result

Segmenting Scenes

We define 0.6 as the default threshold for ECR. The mean ECR is a value around .25 - .30, which means that we should shoot for a threshold way above this value. Otherwise, we would obtain way too many false positives. We verify this by setting 4 different levels of thresholds, .1, .4, .6, .8, to narrow down what an ideal threshold looks like. At .1, the slightest of movements would trigger a new scene point, so every frame essentially qualified as its own contained scene. .4 was slightly better in that it accounted for longer scenes but would have a lot of false positives within the scenes themselves. .8 was too high of a threshold where we could observe thousands of frames and scene shifts passing before a scene is actually flagged. From these general observations, .6 seems like a balanced threshold that doesn't trigger too many false positives or too many false negatives.

Qualitative Analysis

The annotated segmentations from the movie repository do not segment the scenes as we expected, which is depicted in Fig 5. As mentioned in the histogram experiment, we believe that the database creators had a different definition of scenes,

that these annotations may be annotations in terms of movie acts, or their frame calculations may be off from ours. Either way, we will not use these scenes for our calculations as there is no clear metric for what they are doing. Since we don't

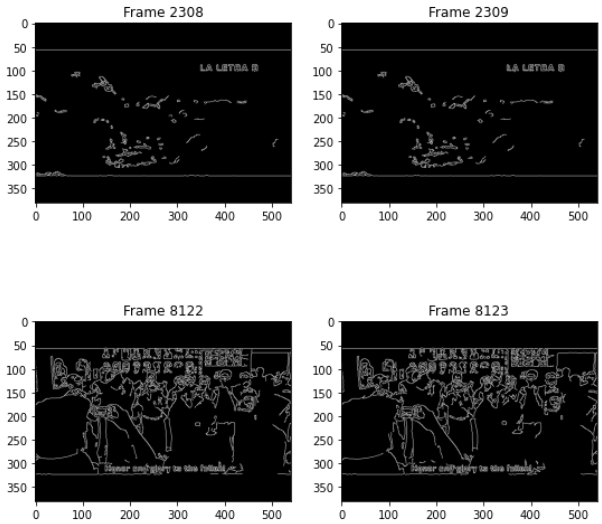


Fig. 5. Annotated Segmentation from Movie

have an 'accurate' label to test against for errors, we will manually identify our own scene segments to test against. The visualization of a scene segmentation using our program is shown in Figure 6. Although, simply testing from the first chunk of scenes in the movie would not be a very good sample as we might encounter a lot of stagnant title shots. To create a better data set to sample from, we need a variety of shots to be tested on such as intra-scene shots that move the camera around, stagnant shots, shots with the subjects moving around, etc. We do so by randomly selecting 15 frames between 0 and

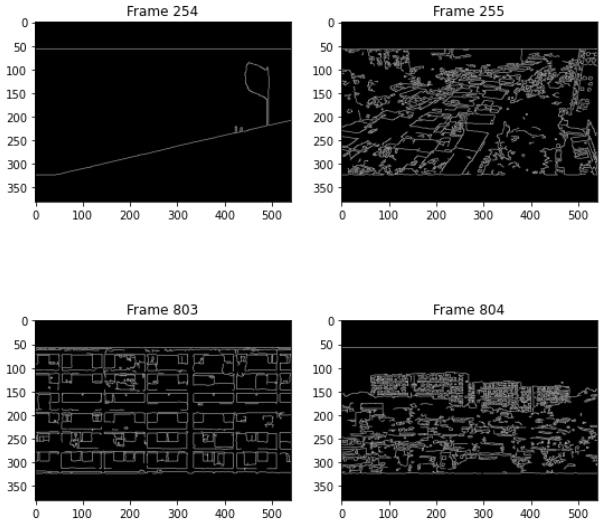


Fig. 6. Initial Scene Segmentation

50,000 frames. With each random frame, we manually scan for the next scene segment and mark that frame down in our 'correct_scenes' array. This value is the start of the next scene segment, so we after running our algorithm, we check if our program correctly identifies the scene as we expect it to. The comparison results of the results are shown in Figure 7.

Clip Name: Fires Beneath Water

Annotations	Random Choosing 15 frames	Was that segment correctly identified ?
[0 , 2308]	4356	Yes
[2309 , 3327]	6239	Yes
[3328 , 4885]	7912	Yes
[4886 , 6084]	9432	Yes
[6085 , 8122]	13866	Yes
[8123 , 9920]	16834	Yes
[9921 , 10490]	24968	Yes
[10491 , 17712]	29521	Yes
[17713 , 18282]	31028	Yes
[18283 , 23586]	32023	Yes
[23587 , 29730]		Yes

Fig. 7. Edge Change Ratio Method Comparison Result

III. Experiment

With the qualitative illustration we provided above, we'll now evaluate our method on a larger scale, which include some of the remaining videos in the dataset.

A. Evaluation metrics

To assess the quality of detected scenes, we take three commonly used and appropriate metrics:

1) The ratio of the number of scenes boundary detected.

2) Miou: a weighted sum of intersection of union of a detected scene boundary with respect to its distance to the closest ground-truth scene boundary. It follows the concept of Jaccard distance between sets and it will be a measure of closeness of our generated annotation sets with tagged annotations. For each ground-truth scene, we take the maximum intersection over union with the detected scenes, averaging them on the whole video. Then the same is done for detected scenes against ground-truth scene, and the two quantities are again averaged. An important note is that both intersection and union are measured in terms of frame lengths for the shots, thus weighting the shots with their relative significance. The final measure is thus given by:

$$M_{iou} = \frac{1}{2}\left(\frac{1}{n}\sum_{i=1}^{n}\max_{j\in\mathbb{N}_m}\frac{\tilde{\mathbf{s}}_i \cap \mathbf{s}_j}{\tilde{\mathbf{s}}_i \cup \tilde{\mathbf{s}}_j} + \frac{1}{m}\sum_{j=1}^{m}\max_{i\in\mathbb{N}_n}\frac{\tilde{\mathbf{s}}_i \cap \mathbf{s}_j}{\tilde{\mathbf{s}}_i \cup \tilde{\mathbf{s}}_j}\right)$$

3) Average Precision with offset: the percentage of annotated scene boundaries which lies within 3s of the predicted boundary. In a traditional movie setting, 3s usually contains 72 frames.

B. Quantitative results

The following three tables showcase our evaluation results for all defined metrics:

Movie	Correlation	Chi-Square	Intersection	Bhattacharyya	ECR
Big Buck Bunny	0.89	0.89	0.89	0.89	1.25
Seven Dead Men	1.00	1.00	0.09	1.00	1.00
Boy Who Never Slept	1.00	1.00	1.00	1.00	1.10
Sintel	0.67	0.67	0.67	0.67	1.44
La Chute Dune Plume	1.00	1.00	1.00	1.00	1.65
Cosmos Laundromat	0.89	0.89	0.89	0.89	2.00
Elephants Dream	1.00	1.00	1.00	1.00	3.60
Jathias Wager	1.00	1.00	1.00	1.00	1.58
Meridian	1.00	1.00	1.00	1.00	2.40
Tears Of Steel	1.00	1.00	1.00	1.00	1.71

Table 1: Scene Boundary Cont Ratio for test movies

Movie	Correlation	Chi-Square	Intersection	Bhattacharyya	ECR
Big Buck Bunny	0.35	0.48	0.38	0.38	0.45
Seven Dead Men	0.38	0.41	0.08	0.42	0.47
Boy Who Never Slept	0.29	0.17	0.10	0.10	0.25
Sintel	0.22	0.41	0.51	0.41	0.53
La Chute Dune Plume	0.32	0.38	0.40	0.40	0.40
Cosmos Laundromat	0.32	0.34	0.37	0.46	0.52
Elephants Dream	0.51	0.38	0.34	0.31	0.49
Jathias Wager	0.27	0.45	0.51	0.52	0.61
Meridian	0.28	0.40	0.42	0.57	0.45
Tears Of Steel	0.35	0.22	0.31	0.31	0.33

Table 2: Average symmetric intersection over union between detected scenes and labeled scenes for test movies

Movie	Correlation	Chi-Square	Intersection	Bhattacharyya	ECR
Big Buck Bunny	0.24	0.59	0.53	0.53	0.77
Seven Dead Men	0.48	0.40	1.00	0.49	0.45
Boy Who Never Slept	0.30	0.43	0.57	0.62	0.65
Sintel	0.40	0.60	0.70	0.60	0.80
La Chute Dune Plume	0.09	0.27	0.36	0.36	0.27
Cosmos Laundromat	0.38	0.50	0.50	0.63	0.58
Elephants Dream	0.40	0.40	0.30	0.30	0.30
Jathias Wager	0.19	0.38	0.38	0.44	0.52
Meridian	0.50	0.50	0.60	0.70	0.60
Tears Of Steel	0.33	0.25	0.58	0.42	0.56

Table 3: Average precision of detected scenes with 3s(72 frames) offset for test movies

IV. Discussion

A. *Histogram Method*

The histogram comparison method is quite useful for segmenting scenes based purely on changes in color/intensity, such as when the next frame's background changes, or when a different character or object is prominently displayed in the following frame. It is also a rather cheap way of performing segmentation, as the bulk of the work required lies in preprocessing and calculating the respective histogram for each individual frame; the actual comparisons of histograms and calculation of the distances to determine which frames make up what scenes are not resource intensive procedures.

On the the other hand, this method can also be viewed as a one-dimensional approach to scene segmentation, as it only takes into account intensity values and nothing more, which may lead to issues with frames where considerable changes in the background, characters or objects occur but are still technically part of the same scene, as seen with the example of the camera shifting between two points of view.

The distance metrics themselves also have their downsides, such as with correlation, which can only represent linear relationships between variables, meaning other relationships get left out. Chi-square distance does not work well for distributions containing many low frequencies, while intersection relies solely on intensity values and treats all bins with equal importance. Finally, Bhattacharyya is similar to intersection but requires a probability distribution.

Overall, each of the distances proved useful for scene segmentation, but as for which one provides the best results, that would need to be determined on a case by case basis. This could be observed in the results obtained during the experimentation phase, as the distance metric which yielded results that were closest to the pre-defined annotations varied from clip to clip.

B. *ECR Method*

The Edge Change Ratio (ECR) is a measure of dissimilarity between different video frames, so by setting a certain threshold, we can theoretically segment scenes by searching for points that have a higher ECR than others. We predicted that this method would be great at detecting scenes that have a hard shift but would perform poorly when it comes to moving scenes.

This hypothesis held true for the most part. As expected, the ECR did a great job at detecting when hard scene changes occurred. This took the form of changing camera views or from changing environments. Each time this occurred, a large dissimilarity would occur between the two frames because there would be a large change in most of the pixel edges which would cause the program to record the point at which the segment ends.

The method also worked well for recognizing moving objects within the scene themselves which was surprising. The ECR looks at the change in edge pixels from one frame to the next, so assuming that objects are moving at a reasonable speed, the ECR will correctly flag the scene. In one example, we observed this when testing on a car parade with multiple moving cars throughout the scene. The same idea can be applied for long single-shot scenes where the camera is actually being moved around. The ECR did relatively well in recognizing a scene where the camera would move throughout an art gallery, however, as mentioned above, this technique only works when objects or the camera move slowly. At the end of this same scene, the camera panned to the side quickly, which triggered the ECR to flag a scene. The objects in the scenes stayed the same but shifted by about 50 pixels each frame, causing the ECR to recognize all of these individual frames as their own scenes.

In addition, when scenes rapidly fade away, chunks of edges disappear entirely. This causes false positives to be fired off because the difference between the scenes is actually high in these cases. While the ECR correctly detected all of the scenes in our randomized frames above, it could be argued that the ECR works a little too well. Edges changes are a decent method for detecting scenes, however it is not robust to some of the issues above. Setting too high of a threshold makes it difficult for some scenes to be detected as expected,

however too low of a threshold and too many scenes will be incorrectly flagged. The threshold we used (.6) worked well but had the issues that were described above.

V. Future work

Histogram and edge based approaches are just two of a wide variety of methods used in segmenting video clips. We considered using object detection through a convolutional neural net to try and segment scenes, however this method proved a little too complex given the topics we covered this semester. In addition, we would have liked to improve our database. This could be by having more data to test on but more importantly, by finding a database that provides scene segments. It was very difficult to manually identify scenes, and given our time constraint, only could opt to test a limited amount of samples. A correctly annotated database would drastically improve our ability to measure our testing error

In addition, we also could have improved upon the methods we implemented. The ECR performed well in detecting camera changes, but was overly sensitive and triggered too many false positives. Because the ECR segmented scenes too well, a future experiment could be done to try and use another method to group back the segmented pieces that actually constitute a correct scene.

The histogram comparison method can be improved by using a more robust distance metric, as the ones used for this paper do not account for cross-bin similarity. One such metric is the Earth Mover's Distance (EMD), which also compares bins within each histogram and determines which ones are better representations of the image, providing for a more comprehensive way to then calculate similarity between frames.

References

[1] Histogram Comparison. OpenCV `https://docs.opencv.org/3.4/d8/dc8/tutorial_histogram_comparison.html`

[2] IBM Dataset: `https://www.research.ibm.com/haifa/projects/imt/video/Video_DataSetTable.shtml`

[3] Mann, J. K., Kaur, N. *Key Frame Extraction from a Video using Edge Change Ratio: Semantic Scholar*. `https://www.semanticscholar.org/paper/Key-FrameExtractionfromaVideousingEdgeChangeMann-Kaur/030ee97e4bf73295bc44a19c54c377d1f17f48ca`, 1970 January 1

[4] Marín-Reyes, P.A., Lorenzo-Navarro, J., Castrillón-Santana, M. *Comparative study of histogram distance measures for re-identification. Instituto Universitario SIANI, Universidad de Las Palmas de Gran Canaria*, 2016.

[5] X. Liu, J. Sun, J. Liu, J. Sun and J. Liu "Shot-based temporally respective frame generation algorithm for video hashing," 2013 IEEE International Workshop on Information Forensics and Security (WIFS), 2013, pp. 109-114, doi: 10.1109/WIFS.2013.6707803.