Movie Genre Classification via Scene Categorization

Howard Zhou
Computational Perception Lab
School Interactive Computing
Georgia Institute of Technology
howardz@cc.gatech.edu

Tucker Hermans
Computational Perception Lab
School Interactive Computing
Georgia Institute of Technology
thermans@cc.gatech.edu

Asmita V. Karandikar
Computational Perception Lab
School Interactive Computing
Georgia Institute of Technology
akarandikar6@cc.gatech.edu

James M. Rehg
Computational Perception Lab
School Interactive Computing
Georgia Institute of Technology
rehg@cc.gatech.edu

ABSTRACT

This paper presents a method for movie genre categorization of movie trailers, based on scene categorization. We view our approach as a step forward from using only low-level visual feature cues, towards the eventual goal of high-level semantic understanding of feature films. Our approach decomposes each trailer into a collection of keyframes through shot boundary analysis. From these keyframes, we use state-of-the-art scene detectors and descriptors to extract features, which are then used for shot categorization via unsupervised learning. This allows us to represent trailers using a bag-of-visual-words (*bovw*) model with shot classes as vocabularies. We approach the genre classification task by mapping *bovw* temporally structured trailer features to four high-level movie genres: action, comedy, drama or horror films. We have conducted experiments on 1239 annotated trailers. Our experimental results demonstrate that exploiting scene structures improves film genre classification compared to using only low-level visual features.

1. INTRODUCTION

Motion-pictures play a significant role in fulfilling people's entertainment needs. Today, thanks to advancements in Internet technology, consumers have access to an unprecedented amount of movies from various on-line services. This has created the need for automatic content-driven movie recommendation. Although considerable advancements have been made in the areas of video retrieval and collaborative filtering [1], movie genres still act as a key attribute in such recommendation systems. Being able to automatically classify movies by genre would enable 1) indexing multimedia databases to help search for particular types of film, 2) automatically identifying movies for consumers through user preference modeling, 3) facilitating automatic movie content filtering and summarization. Therefore, over the years, researchers have proposed various automatic genre classification methods for both movies [8, 2, 4] and general video data [3, 9, 6, 11], exploring various video features such as noisy text labels [11], closed captions [2], audio features [3, 9, 7], and low-level visual features [3, 2, 8, 4].

In this paper, we address the specific problem of genre classification of cinematic trailers. We choose to work with trailers for three reasons: First, a trailer can be viewed as a concise "summarization" of an entire movie. Often considerable effort is spent in producing a trailer in order to market the associated movie effectively. Second, trailers are easily obtainable in large quantities from the Internet. Third, trailers require much less storage and processing resources in comparison to full-length movies. The problem of trailer genre classification has received some previous attention. Notably, Rasheed, et.al. [8] proposed using four low-level visual features, average shot length, color variance, key-lighting, and motion content, in order to classify trailers into four genres: action, comedy, drama, and horror. They showed that these features, which were inspired by cinematic practices, led to a good genre classification performance on a set of 100 Hollywood movie trailers. One of the contributions of this work is to revisit these features for a much larger trailer dataset. We find that these features are much less discriminating when the database includes more than a thousand trailers (see Figure 5(b)).

Our approach to genre categorization is based on the hypothesis that scene categorization methods can be applied to a collection of temporally-ordered static key frames to yield an effective feature representation for classification. We explore this assumption by constructing such an intermediate representation of movie trailers. In our method, we decompose each trailer into a series of shots and perform scene categorization using state-of-the-art scene feature detectors and descriptors. These automatically-learned scene classes are then used as the "vocabulary" of movie trailers. Using the *bag of visual words* (*bovw*) model, each trailer can be represented as a temporally segmented 2D histogram of scene categories, which allows the calculation of trailer similarities. We collect a large database of movie trailers from the Internet to build such a scene vocabulary, and then use their ground-truth genre labels to infer the genre of a new,

MM'10, October 25–29, 2010, Firenze, Italy.

unseen trailer. This paper makes three contributions: **1)** a novel automatic genre classification method that utilizes state-of-the-art scene categorization methods. **2)** a similarity measure between trailers that incorporates an intermediate level of scene representation and temporal structure. **3)** a large database of movie trailers with genre labels that we will make publicly-available.

2. SHOT BOUNDARY DETECTION

The first step of our approach decomposes a trailer into a series of shots using the shot boundary detection algorithm described in [8]. A trailer is first converted into its n frames. For each frame i, we generate a histogram H_i of its HSV color space representation, with bin dimension 8, 4, and 4 for the hue, saturation, and value components, respectively. For every consecutive frame pair, we compute the intersection of the histogram $S(i) = \sum_{j \in \text{all bins}} min(H_i(j), H_{i-1}(j))$, where j is used to index histogram bins. S is further smoothed iteratively using a Gaussian kernel with variance proportional to the signal gradient, eliminating erroneous local minima. In the end, shot boundaries are detected where two consecutive frames have a local minimum in S.

We use a single keyframe as a static representation of a shot. To select this keyframe, we wish to exclude frames that are near and on shot boundaries because they are likely to contain undesirable transition artifacts. Sophisticated keyframe extraction algorithms exist; however, we find selecting the middle frame of a shot to be sufficient.

3. SCENE CATEGORIZATION

The shot boundary detection step converts a set of trailers t_i into a collection of shot keyframes k_{ij}, where i is the trailer index and j is the shot sequence index. The scene features from keyframes can now be analyzed using several state-of-the-art feature detectors and descriptors. In this paper, we choose GIST [5], CENTRIST [12], and a variant which we call W-CENTRIST. We first briefly describe each feature descriptor. We then explain how we map these features to scene category labels and finally show how these intermediate level features are used to perform the genre classification task.

3.1 GIST

The GIST model produces a single, holistic feature descriptor for a given image, which attempts to encode semantic information describing characteristics of the image scene [5]. It has been shown to perform well in semantic scene classification of images. The semantic properties of interest describe high level characteristics of the scene, such as its roughness, ruggedness, and openness. Spectral analysis methods compute features that encode these properties over windows in the image, while preserving their spatial layout in the scene.

3.2 CENTRIST

CENTRIST, the CENsus TRansform hISTogram, is a visual descriptor developed for recognizing the semantic category of natural scenes and indoor environments, e.g. forests, coasts, streets, bedrooms, living rooms, etc. It has been shown that CENTRIST produces outstanding results for the place and scene recognition task when used within the *bovw* framework [12]. To compute CENTRIST features, an image first undergoes a non-parametric local transform called

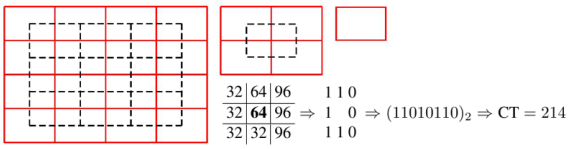

Figure 1: Illustration of the level 2, 1, and 0 split of an image (left) and Census Transform (right).

Census Transform (CT) which establishes correspondence between local patches. The Census Transform compares the intensity values of a given pixel with its eight neighboring pixels as shown in Figure 1(right) and generates an eight bit CT value. Since the Census Transform captures local structures while retaining global structures, a histogram of CT values in an image patch is shown to encode both local and global information of the image. To generate CENTRIST features, a spatial pyramid (as seen in Figure 1) is employed to capture spatial structure at multiple scales. Concatenating all the CT histograms calculated within all red and black blocks of Figure 1 produces the final descriptor.

3.3 W-CENTRIST

Both GIST and CENTRIST models discard color information when generating descriptors. However, we think color plays an important role in conveying the mood of a scene. Therefore, we have devised W-CENTRIST, a variant of the CENTRIST descriptor that captures color information in addition to intensity statistics. This descriptor is constructed in the W invariant color space, a derivation of the opponent color space where $W_1 = \frac{O_1}{O_3}$ and $W_2 = \frac{O_2}{O_3}$. Channel O_1 and O_2 store the color information, and the division by intensity channel O_3 makes this statistic intensity-invariant. Color descriptors constructed from the W invariant color space have been shown to out-perform descriptors formed from the RGB, HSV and Opponent color-spaces [10]. To form the W-CENTRIST descriptor, we extract the CENTRIST features for both W channels and concatenate them into a single feature vector.

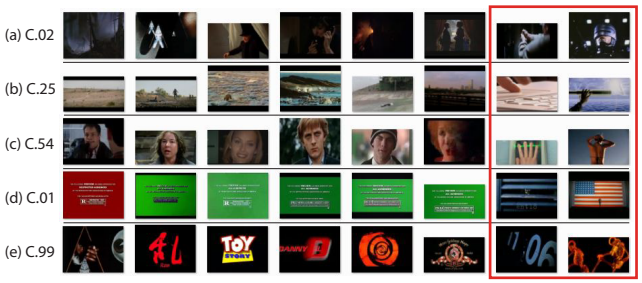

Figure 2: Example shot categorizes generated from the W-CENTRIST clustering (cluster 2, 25, 54, 1, 99 top to bottom).

3.4 Shot Clustering and Trailer Features

For any given feature descriptor, we perform the following procedure to generate feature vector representations for each trailer. First, we generate a scene descriptor for each shot keyframe k_{ij}. We then vector-quantize these scene descriptors using a K-entry visual word codebook, which is obtained from running the K-means clustering algorithm on

a randomly sampled subset of the descriptors. After quantization, each shot is associated with a discrete visual word index $c_j \in \{1, \ldots, K\}$. Figure 2 shows example keyframes from several shot classes using W-CENTRIST features and K set to 100.

We show non-canonical examples of each category in the last two columns of Figure 2. For example cluster 25 is mostly open spaces, but a closeup of a human typing on a keyboard and a logo from a film company also appear as members. Within each shot class, a majority of images (columns 1-6) tend to correspond to a common scene category, such as a dark moody setting (cluster 2), an open space (cluster 25), a frontal face (cluster 54), strips of text on a uniform background (cluster 1), and a type of logo on a dark background (cluster 99). Although outliers like these do exist, we feel that the clustering has already reached the desired precision for this level of scene categorization.

Given the shot codebook, each trailer can be represented by the *bovw* model as a histogram of the shot classes that appear within it. To account for the tempo of the trailer, we weight each shot with its corresponding shot length when constructing the trailer feature vector. In fact, with the introduction of W-CENTRIST to both capture scene intensity and color statistics, we have incorporated the low level visual features, shot length and color variance, into our approach in a holistic manner. While this feature captures the scene content of the trailer, it overlooks the temporal structure of the shots. In order to incorporate the temporal structure, we build a 2D histogram where we bin shot counts first by relative time segment in the trailer (i.e. first third, last half) and then in the shot category histogram as above. This formulation creates another free parameter which examine in Section 4. In the following section, we show empirical evidence of the usefulness of our approach in genre classification.

4. EXPERIMENTS

We have collected 1239 movie trailers from various websites: IMDB (Internet Movie Data Base)[1], Apple Trailers[2], AllTrailers.net[3], and Movie-List.com[4], etc. The corresponding genre information for each trailer is automatically extracted from IMDB. The IMDB website classifies most movies in one to three genres from a total of twenty-four possible genres. We are interested in performing genre classification on the four most frequent genres: action, comedy, drama, and horror. There are 317 trailers that are associated with the action genre, 652 comedy, 1302 drama, and 296 horror. 228 trailers have one associated genre, 340 have two, and 671 have three. Most of the trailers run between one and three minutes; we decode the video tracks and analyze them at a frame rate of 24 fps. The entire database produces around 4.5 million frames. Notice that our preprocessing step does not remove frames containing trailer artifacts such as copyright warnings or movie credits (Fig. 2 d and e). Instead, we rely on the shot clustering step to detect and group keyframes corresponding to these artifacts. The next step in the pipeline detects shot boundaries and extracts the middle frame within each shot as the keyframe, producing approximately 120,000 keyframes in total.

[1]http://www.imdb.com
[2]http://www.apple.com/trailers
[3]http://www.alltrailers.net
[4]http://www.movie-list.com

We extract features from all keyframes using the three feature representations: GIST, CENTRIST, and W-CENTRIST. For GIST features, we used the code described in [5], which requires us to scale our keyframes to 320×320 pixels. GIST descriptors are extracted for each keyframe using four windows at three scales resulting in a descriptor of size 960. For both CENTRIST and W-CENTRIST features, we rescale keyframes to a width of 320 pixels, preserving the aspect ratio, and extract features using the code provided by the authors of [12]. We use 200 codewords and three split levels for CENTRIST features, and 200 codewords and two split levels on both W-invariant channels of the keyframe for W-CENTRIST features. The descriptor sizes are 6200 and 2400 respectively. We randomly sample a subset of $20,000$ features from all trailers and perform K-means clustering to obtain K feature clusters. Since we do not know how many shot classes naturally exist, we run K-means at varying levels of K from 50 to 4000 doubling K at each step. Our results indicate setting $K = 100$ leads to the best results for all three features, so we represent trailers as a 100 dimensional histogram of shot classes, weighted by corresponding shot length. We create T of these 100 bin histograms depending on the chosen temporal resolution. We examine resolutions ranging from 1 to 5 bins finding the peak performance at $T = 3$. To calculate the similarity between two trailers we use histogram intersection, correlation, or χ-square distance measure. Our informal experiments suggest that the χ-square distance performed slightly better and is thus used for the experiments in this section.

The histogram distance gives us a similarity measure among trailers, which we can use to identify the N nearest neighbors for each trailer. Figure 3 shows several examples from the returns of the N nearest neighbor search ($N = 5$). In each example, the first row displays the movie posters corresponding to the query trailer and its nearest neighbors; the second row shows their corresponding ground-truth genres. We also take into account the order of the genres listed on IMDB for each movie. We set weights of 1.0, 0.8, and 0.6 for each of the genres respectively. This corresponds to the bar height in the ground-truth panels. The third row shows the likelihood of each movie belonging to one of the four genres calculated by our method.

Quantitatively, Table 4 shows our genre classification accuracy using a shot vocabulary of size $K = 100$ and χ-square distance as the trailer descriptor similarity measure. For each of the scene feature descriptors, we vary the number of nearest neighbors N between 3 and 5, and the number of temporal bins $T = 1, 3, 5$. The accuracy is computed by counting the number of correctly assigned genre labels. Matches are recorded when the trailer's genre likelihood reaches a threshold of 0.33 and matches the ground truth label. Otherwise, it is considered a mismatch. We also compare our results with the accuracy achieved by using the low level visual features introduced in [8]. We note that in [8], the authors group 100 trailers into six modes identified by Mean-shift clustering and label each clusters according to the dominant genre of the cluster members. This method is not directly comparable to our scheme. Therefore, we apply our K-nearest neighbor search procedure using the four features calculated according to [8] for comparison. Our results show a clear improvement. Figure 5(c) illustrates one of the short-comings of relying solely on low level visual features. There are often large overlaps between the distribution of

Figure 3: Example returns from the nearest neighbor search. From left to right, column one shows the query, and columns 2-6 show nearest neighbor returns. The first row displays movie posters. The second row shows their corresponding groundtruth genres, where red indicates action, green comedy, violet drama, and blue horror. The panel on the third row is the estimated genre likelihood for the query trailer.

these features across genres. However, we believe that our content based scene categorization method captures information which is more indicative of the movie genre. We also show the confusion matrix of matching percentage for genre classification using CENTRIST with $K = 100$ and $N = 5$, and $T = 3$ in Figure 5(a).

Additionally, we perform the same genre classification procedure on 144 new trailers not included in the training set, using the same shot categorization labels from the original 1239 dataset and the best parameter settings for the original experiment: $T = 3$, $N = 5$, and χ-square distance measure. This time, the overall accuracy is 71.58%, which is slightly worse than the previous result. This is expected since no shot from the new trailers is used to generate the codebook, and nearest neighbors are only selected among the new trailers. Nevertheless, these findings suggest that this approach generalizes to unseen trailers.

	N = 3			N = 5		
T	1	3	5	1	3	5
GIST	69.5	68.6	69.5	71.8	71.2	71.6
CENTRIST	72.5	73.0	71.0	74.1	**74.7**	73.6
W-CENTRIST	70.3	69.8	71.7	73.1	72.4	73.2
Low-level	64.3			65.0		

Figure 4: Genre classification accuracy computed using a shot vocabulary of size $K = 100$ and χ-square distance as trailer descriptor similarity measure.

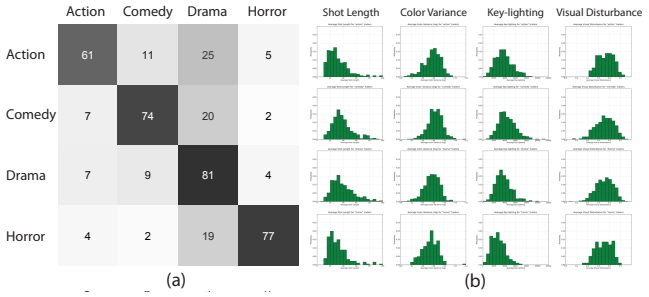

Figure 5: (a) Confusion matrix (K = 100, N = 5, $T = 3$, χ-square distance). (b) Distribution of trailers against low-level features proposed in [8]

5. DISCUSSION AND CONCLUSION

Our experiments demonstrate the usefulness of introducing scene features for movie genre prediction. However, our results leave room for improvement. First, our method does not consider the dynamic components of scenes, since it lacks a representation of the action and movement within shots. Second, although we constructed our database such that erroneous entries are all removed, noise still exists. The trailers have a wide variety of aspect ratios and resolutions, making direct feature comparison difficult; additional compression artifacts within some videos significantly distort the extracted scene features. Lastly, some movies were filmed in black and white, making their scene features quite different from those present in the modern day trailers.

We have presented a framework for automatic classification of film genres using features from scene analysis. We have demonstrated that a temporally-structured feature based on this intermediate level representation of scenes can help improve the classification performance over the use of low-level visual features alone. In the future, we will build upon our static scene analysis to include scene dynamics, such as action recognition and camera movement estimation, to help achieve higher-level dynamic scene understanding.

6. REFERENCES

[1] R. Bell, Y. Koren, and C. Volinsky. The BellKor 2008 solution to the Netflix Prize, 2008.

[2] D. Brezeale and D. J. Cook. Using closed captions and visual features to classify movies by genre. In *7th Intl. Workshop on Multimedia Data Mining*, 2006.

[3] S. Fischer, R. Lienhart, and W. Effelsberg. Automatic recognition of film genres. In *ACM Intl. Conf. on Multimedia*, pages 295–304, 1995.

[4] H.-Y. Huang, W.-S. Shih, and W.-H. Hsu. A film classifier based on low-level visual features. *Journal of Multimedia*, 3(3):26–33, Jul 2008.

[5] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Intl. J. of Computer Vision*, 42(3):145–175, May 2001.

[6] C. Ramachandran, R. Malik, X. Jin, J. Gao, K. Nahrstedt, and J. Han. Videomule: a consensus learning approach to multi-label classification from noisy user-generated videos. In *ACM Intl. Conf. on Multimedia*, 2009.

[7] Z. Rasheed and M. Shah. Movie genre classification by exploiting audio-visual features of previews. In *Proc. of Intl. Conf. on Pattern Recognition (ICPR)*, 2002.

[8] Z. Rasheed, Y. Sheikh, and M. Shah. On the use of computable features for film classification. *IEEE Trans. Circuits and Systems for Video Technology*, 15:52–64, 2003.

[9] M. Roach and J. Mason. Classification of video genre using audio. In *Proc. Eurospeech*, pages 2693–2696, 2001.

[10] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluation of color descriptors for object and scene recognition. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[11] Z. Wang, M. Zhao, Y. Song, S. Kumar, and B. Li. Youtubecat: Learning to categorize wild web videos. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[12] J. Wu and J. M. Rehg. Where am I: Place instance and category recognition using spatial PACT. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008.