Dancer Identification within Detected Shot Boundaries

Kedar Reddy
Department of Visual Arts
University of California San Diego
San Diego, CA 92122
kreddy@ucsd.edu

Nevin Gaudreau
Department of Computer Science
University of California San Diego
San Diego, CA 92122
ngaudrea@ucsd.edu

Nishanth Satyanarayana
Department of Electrical Engineering
University of California San Diego
San Diego, CA 92122
nsatyana@ucsd.edu

Abstract

This project began with an aim to develop tools to categorize identities and attributes of dancers and their surroundings from song-dance sequences of Telugu cinema. We developed a framework, Dance-GanceX, in C++ which heavily relies on OpenCV to track such values as the amount of movement, color histograms, and intensities per dancer over time. Shot boundary detection and Paint-Trails(camera stabilization) are the other features of Dance-GanceX used to pre-segment the song-dance videos. Dance-GanceX will facilitate better understanding of the culturally influential song-dance sequences that define popular Indian Cinema, providing critics, commentators, and artists a reliable tool to extract meaningful data and hypothesize from.

1 Introduction

Idlebrain.com's reviewer, Jeevi analyzing the film Arya2 asserts: "Allu Arjun is probably the best dancer of current era in Tollywood. That is the reason why he made hugely difficult dances appear fluid and effortless in the first four songs of the movie."Critics such as Jeevi have been around since the advent of cinema. Their jobs have been to analyze the large volumes of media being produced and make the public aware of their quality/merits. Everyone form amateur Youtube reviewers to professional critics influence our sense of what is good and bad resulting in a loss of objectivity. About three months ago, towards the tail-end of 2009, the three of us ceased to argue about who was the better dancer between Prabhu Deva, Jr.NTR, and Chiranjeevi and decided to back up our arguments with numbers. We were going to quantify such heretofore abstract terms as limb movement, and synchronization between dancers. Thus, the idea for Dance-GanceX took shape. Today Dance-GanceX is a reality and able to segment dance videos according to shot-boundaries and output data considering each dancer's movement and attire for further analysis.

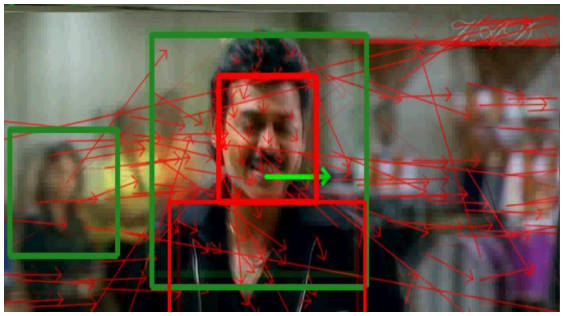

Figure 1. Optical flow vectors and human detectors firing on a dance video frame.

2 Approach

Owing to its great compatibility with Computer Vision, we will be making use of OpenCV and its various libraries to write our code [2]. As our primary aim in the project is to analyze dance videos quantitatively, we first perform frame-by-frame as well as shot-by-shot analysis of the videos. Within each shot, we aim to extract the human count under the assumption that every human in the music video is also a dancer. The results will contain number of dancers in the shot accompanied by character descriptors of the dancers. The descriptors include color information as well as dimensions including height and width of the dancer.

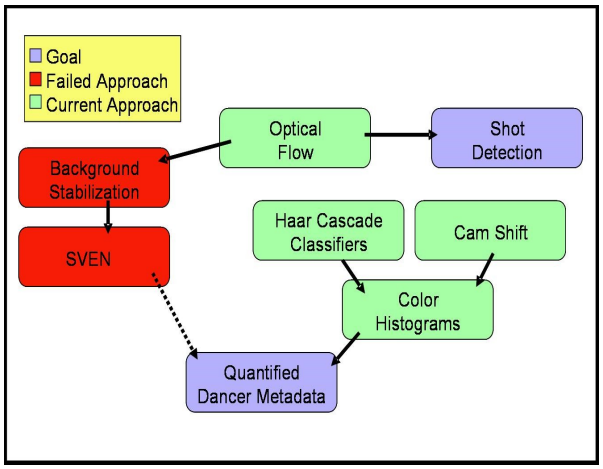

Figure 2. A flow-chart that depicts the approach

Our initial approach was to divide the dance videos into segments along shot boundaries. These segmented videos were then meant to be inputted into SVEN [6,7], an open-source, real-time, pedestrian tracker. However, SVEN required the background scene to be static at all times. Obviously with dance videos, the background is very rarely static, especially with color-cinema's complex cinematography. To tackle this problem, we implemented camera stabilization on the dance videos to create an illusion of a static background. Even after inputting our post-processed videos into SVEN, we did not get satisfactory results in terms of detecting dancers. We were forced to change our approach.

Our current approach involves implementing human detection using Haar cascade classifiers [5]. Every time the classifiers fire, information such as human body dimensions as well as color information about the area where the classifier fired are extracted.

2.1 Camera Stabilization

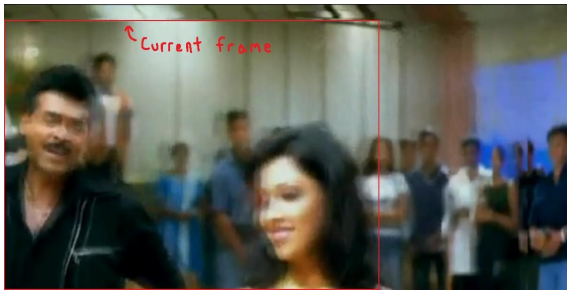

Figure 3. A screen-shot showing camera stabilization.

The goal of camera stabilization is to make the background look less "jerky." This is a two-step process. First, we implement D. Staven's optical flow [1] on our video sequences, in which one frame is compared with the frame immediately following it. We pick features in the current frame and track its movement in the next frame. We track 400 features. We would like to have as many features to track as possible. However, adding more features to track considerably slows down our processing. We found 400 to be the right balance where it allows us to run our code at a reasonable pace while at the same time retaining enough features to get satisfactory optical flow information. We are left with an array of size 400 which describes how much each of the 400 pixels moved from one frame to the other. We acquire the median optical flow vector to describe the motion of the features from one frame to the other.

Using this median optical flow vector, we compensate the movement of the background by manually shifting our picture by the same amount as the median optical flow vector in the opposite direction of the median optical flow vector. This will create a mosaic type effect as seen in the image below, where an illusion is preserved that the background is static.

For this to work successfully, it is required that more than 50% of the frame is background. Picking the median optical flow vector if more than 50% of the frame is background ensures that only information about the background is retained.

Although we are no longer implementing camera stabilization, it is worth while to mention its importance to our project's progress. In the process

of stabilizing our dance videos using optical flow, we stumbled across a very useful application of optical flow: Shot Detection

2.2 Shot Detection

We are interested in implementing shot detection because analyzing video sequences on a per-shot basis makes it easier. There are various elements of a dance sequence that may change from shot to shot including number of dancers in the video, location, etc. However, the theme throughout the course of the song will tend to remain constant. Our aim is to figure out what this trend is by analyzing the dance sequence on a shot-by-shot basis.

We will use to our advantage the fact that certain characteristics of a video sequence change between shots. Pixels tend to change dramatically between shots. So we will use the same idea that we used in our camera stabilization implementation. We will extract optical flow vectors between each consecutive frames. If the median optical flow vector blows up, then that is an indication of a shot boundary. The optical flow vector blows up, because the algorithm is trying to find the same pixels in a different shot, and it is not able to since those pixels are not present within the frame anymore.

Figure 4. A visual depicting shot boundary detection, with median optical-flow coordinates per frame.

2.3 Human Detection

We are using Head and Shoulders [8] Haar Cascade Classifier to run face detection on a frame-by-frame basis. The Classifier draws a box around the face and shoulders. Using knowledge of human-body proportions, we extend this box down to the lower body. We then extract color information within these individual boxes. Based on the dimensions of the box and the color information within these boxes, we can predict other information from the videos.

Figure 5. A screen-shot of different Haar cascade classifiers(red and green boxes) and the cam-shift(purple ellipse) tracker firing.

In Indian cinema, Telugu cinema in particular, we see a trend where the lead dancer is wearing clothes different from the background dancers, while the background dancers tend to wear similar clothing. We will take advantage of this trend that we see, and extracting color information on the boxes drawn above, we can differentiate between lead and background dancers. Another thing that will help us differentiate between background and foreground dancers is their position in the frame. If they are in the background, then relatively they will appear smaller than the foreground or lead dancer. The dimensions of the boxes (in pixels) will help us identify the dancers' roles in the frame. We extract these “boxes” by setting ROI on the original frame [3,4].

3 Data and Results

We found that the shot detection portion of our program has about 20% error. The detection is robust when there are inter-shot transitions that are simple direct cuts. However, when the transition is of a dissolving or fading nature, the shots are detected only 20% to 30% of the time.

The results of the camera stabilization portion of our program were satisfactory because of our use of optical flow vectors. For sections in videos that had simple camera translation, the camera stabilization was optimal because of the lack of camera rotation and zooming, which we did not care to worry about. With the help of the frame paint trails that were being drawn, the previous scenes stay visibly constructed as the video continues.

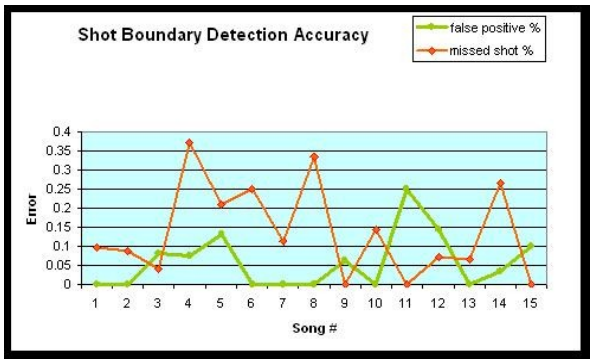

Figure 6. Graph of the accuracy of the optical-flow shot-boundary detector over 15 dance videos.

In terms of actually detecting the dancer's face and body, we have obtained over ten different Haar Cascade Classifiers that try to locate various characteristics that match what a face or body looks like. After trying multiple cascades, we realized that using more than one cascade severely decreases performance because of the time it takes to go through multiple cascades every frame. After much testing with the various cascades and different videos, we concluded that the 'head
and shoulders' cascade works the best by itself. This cascade has an accuracy percentage of about 65 to 75.

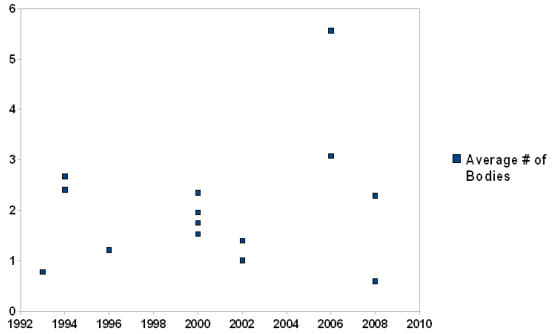

Figure 7. Graph summarizing human detection in 15 dance videos from the 1990 to 2010.

4 Future Work

We hope to streamline Dance-GanceX into a batch pipe-line that will be able to work with any number of videos and output xml files with critical information about color, movement per dancer, frame, and shot for each video. We also intend to add functionality to Dance-GanceX that will be able to quantify the amount of dance per shot based visual patterns within a frame and across time. Eventually we hope for Dance-GanceX to output compelling visualizations along with raw data.

5 Acknowledgments

Thanks to Professor Serge Belongie for helpful advice throughout the course. Also, thanks to the rest of the CSE 190 classmates for inspiration and guidance.

References

[1] D. Stavens. *Sparse Optical Flow Demo Program*, 2007. Available from: <http://robotics.stanford.edu/~dstavens/cs223b/>.

[2] R. Laganière. "Laganiere's Tutorial on OpenCV and VC++." VIVA lab, University of Ottawa, 2003.

<http://www.site.uottawa.ca/~laganier/tutorial/opencv+directshow/cvision.htm>.

[3] A. Nashruddin. "OpenCV Region of Interest (ROI)." *Image Processing & Computer Vision with OpenCV*, 2009.

<http://www.nashruddin.com/OpenCV_Region_of_Interest_(ROI)>.

[4] Z. Lin. "OpenCV -- 5: handling the subimage by ROI function." *On the way to the Virtual World. Create the World as we want,* 2009.

<http://www.zhongshanlin.com/2009/09/opencv-5-handling-subimage-by-roi.html>.

[5] Q. Zhu, S. Avidan, M. Ye and K-T Cheng, Fast human detection using a cascade of Histograms of Oriented Gradients, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* NY, NY, USA, 2006.

[6] V. Rabaud. SVEN: Surveillance Video Entertainment Network (aka AI to the People), 2005-2006.

Available from: <http://deprogramming.us/sven/>.

[7] A. Alexander, About... Software, Surveillance, Scariness, Subjectivity (and SVEN), In *Transdisciplinary Digital Art: Sound, Vision and the New Screen*, pages 467-475, Springer, 2008.

[8] M. Castrillon Santana, O. Deniz Suarez, M. Hernandez Tejera, C. Guerra Artal. ENCARA2: Real-time Detection of Multiple Faces at Different Resolutions in Video Streams, In *Journal of Visual Communication and Image Representation*, pages 130-140, 2007.