

3DMP: The 3D Movie Player

Kishor P. Deshmukh & Gauri K. Bodkhe

Abstract - The main objective of this document is to present the detailed description of the “3DMP: The 3D Movie Player”. This application lets you view 3D videos on the go, using your android mobile devices. Which makes it easy for viewing of the 3D videos anywhere, anytime with lot of ease. Our project is salutary in a way that you can view 3D videos with a better ease by using any home-made 3D cardboard viewer for android. Google has recently came up with this Cardboard I/O making it easier for developers to create applications for 3D SBS (3D Side-By-Side) viewing and do the testing for same. Thus, thereby bringing the pleasure of immersive 3D environment to even common man.

Keywords – Android, Smart Phones, 3D, immersive, easy video.

I. INTRODUCTION

The perspective of this application is to provide the easiness to control the 3D immersive SBS movies or videos as well ease them to a better extent to eschew the head-aches and dizziness caused by it during the playtime of SBS video. According to the survey it has been noticed that it is very cheap to prepare the virtual reality goggles/headset. The cost merely ranges from \$7~20(apart from mobile) to prepare one, while comparing it with real virtual reality headset, the cheapest available for \$300~1800+. Of course the immersion's and depth's have a great variance. This project is about playing and controlling of the 3d videos by using the onscreen controls as well as gestures. Currently only supporting the android smart phone/tablet, but soon will be published for other operating systems too.

Literature Survey of the Project:

1. 3DVPLAYER :

- Plays 3D movies on only androids with actual 3D no glass display.
- Available in Chinese language.
- Incorrect aspect ratio for most videos having disparate aspect ratio.
- No per-frame separation.

2. Eassee3D:

- Another no glass 3D video player.

-Promises quality 3d effects but requires a sheet of filtering material to be placed over screen.

Manuscript received Dec, 2014.

K. Deshmukh finished studying in Govt. College Of Engineering And Research, Awasari(kd), Pune.

G. Bodkhe finished studying in Govt. College Of Engineering And Research, Awasari(kd), Pune.

-3D video/images/games appear little dark as compared to original brightness.

-Filtering material available for most of the Android and Apple devices.

3. 3D video players have already started, then why 3DMP?

The primary differences between the two disciplines are the distinction between availability of the features required to ease 3D viewing of videos and images. Despite of the presence of such 3D video players, there was need for 3DMP, to ease the viewing of 3D, using comfortable controls and gestures.

II. TRADITIONAL WAYS OF PLAYING AND VIEWING 3D VIDEOS

There are two types 3D SBS movies: the normal SBS having the left image for left eye and the right for right, while other is cross-eyed. The cross-eyed requires no glasses for viewing but is a great headache for viewing and perceiving it as 3D. Although the traditional normal SBS 3D can also be viewed in the cross-eye fashion, by stressing the eyes while requiring no glasses, too causes even more headache. For e.g.,

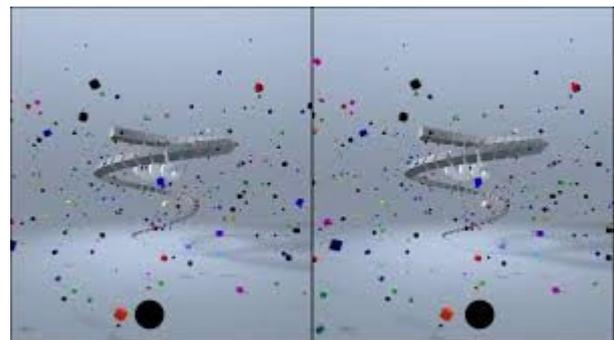


Fig. 1. No glass 3D cross eyed movie from YouTube.



Fig. 2. Actual 3D SBS video from YouTube.

Both the videos are readily available on YouTube as well as other famous video-websites. But the cross-eyed videos cannot be viewed as 3D in the android virtual headset.

III. DRAWBACKS IN THE CURRENT VIDEO PLAYERS

The current video players which support the 3d viewing, by either processing or without processing the videos, have multiple drawbacks, which 3DMP overcomes and will be discussed below. The aspect ratio and spacing between the 3d frames cause the dizziness and often headaches. The 3D SBS videos mostly those, having the horizontal width very less fall into this category. This type of videos make the viewer strain his eyes into the center and the brain simultaneously tries to merge this two image into one in order to form a single image which also is half the total pre-processed video.

e.g.

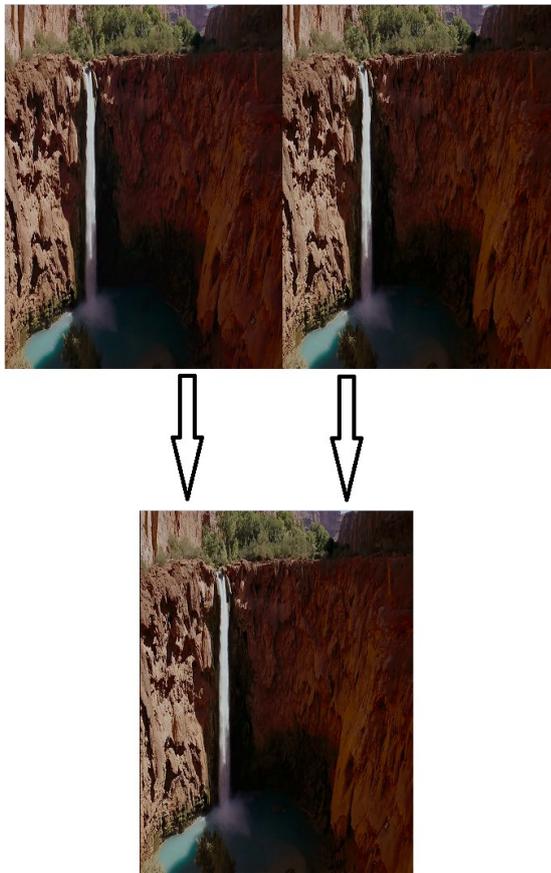


Fig. 3. SBS Video having very less width.

From this image one can understand that, how less width in actual video matters the overall output width of real video. For this type of image having very less or almost no gap in between the images make the viewer concentrate on the center of the screen thus straining the eyes.

The problem can be easily understood by the following figure where, eyes are shown while viewing of the 3D SBS movie.

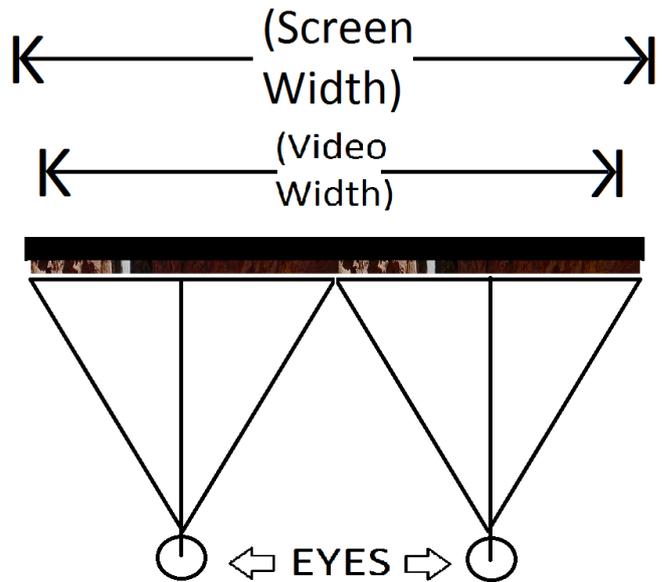


Fig. 4. 3D SBS video with normal width with respect to display width.

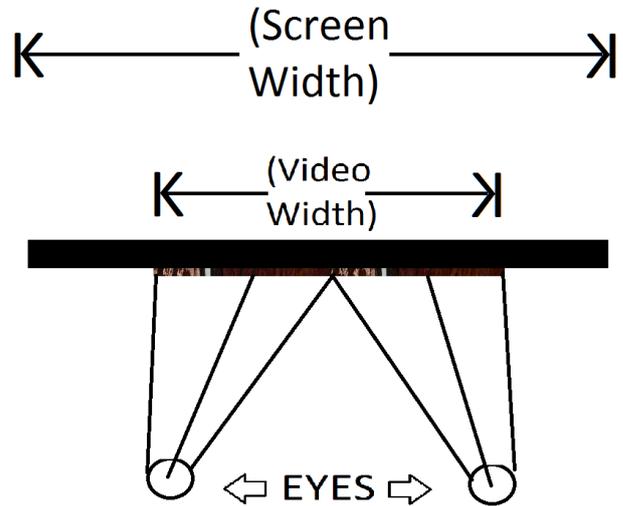


Fig. 4. 3D SBS video with stinted width with respect to display width.

Another drawback is the one with economical home-made videos, which are generally taken using the pair of two cameras. While combining these two videos from different cameras, rarely the condition appears, where the frames of one side are skipped to a few micro-seconds. This usually occurs when the videos are manually captured or/and further combined manually. As, both the images placed are not in synchronous, the brain is never able to merge this two images into one. But, the brain is continuously trying to merge these images, the user starts to get headache. One such snapshot of a video from YouTube.



Fig. 5. 3D SBS video skipped frames.

Furthermore, the controls and sub-titles for the videos are overlaid on the whole SBS video. This is very irksome as, the text and controls viewed from the headset are ambiguous to perceive during each video play. One such screen-shot of 3D SBS movie player, where texts and controls overlap, is given below.



Fig. 6. Output of 3D SBS video having controls and text on both sides.

IV. PROPOSED SYSTEM:

For playing of the 3d SBS video, we need two different pictures on the screen at same time. Thus for this objective we proposed the application to be a 3D Movie player, wherein we will need to take into account the screen is divided into half, one half containing first image and other half containing the other image.

For this purpose we took two textureVideoView's by "Sprylab Technologies" in the application in correct proportions, but taking into account that it stays full-screen, so that notification bar gets omitted. And upon touching the options button only, the notification bar will appear.

During the play-time of video, two instances of same video are played on the screen, while muting one of them. Each texture-video-view plays one instance of video. The purpose of using the texture-video-views is to crop the video in two halves. The texture-video-view of left side plays the left half of the SBS videos while, the right plays the other half of the cropped video. This is done to achieve a large advantage, which will be discussed further. The code for the implementation of which is given below,

```
private void updateTextureViewSize(int viewWidth, int
viewHeight, int side) {
    float scaleX = 1.0f;
    float scaleY = 1.0f;
```

```
//checking the screen-size for setting appr. aspect-ratio
if (moblieVideoWidth > viewWidth && mVideoHeight >
viewHeight) {
    scaleX = moblieVideoWidth / viewWidth;
    scaleY = mVideoHeight / viewHeight;
} else if (moblieVideoWidth < viewWidth &&
mVideoHeight < viewHeight) {
    scaleY = viewWidth / moblieVideoWidth;
    scaleX = viewHeight / mVideoHeight;
} else if (viewWidth > moblieVideoWidth) {
    scaleY = (viewWidth / moblieVideoWidth) / (viewHeight
/ mVideoHeight);
} else if (viewHeight > mVideoHeight) {
    scaleX = (viewHeight / mVideoHeight) / (viewWidth /
moblieVideoWidth);
}
```

```
// Calculate pivot points, in our case crop from center
int pivotPointX = viewWidth / 2;
if(side==0)
    int pivotPointY = 0;
else
    int pivotPointY = moblieVideoWidth;
```

```
Matrix matrix = new Matrix();
matrix.setScale(scaleX, scaleY, pivotPointX, pivotPointY);

mTextureView.setTransform(matrix);
mTextureView.setLayoutParams(new
FrameLayout.LayoutParams(viewWidth, viewHeight));
}
```

Here the size of the video is passed to this function for appropriate cropping. And the videos are then perfectly

divided into half. Thus, thereby, playing the two instances of same video.

V. SOLUTION TO THE DRAWBACKS:

Thus, for the spacing between the right and left images of the video we set the cropped video to the center of the "texture-video-views". This does not cause the strain on the eyes and a user can see the video with relaxed eyes. This is the explanation through diagram, of what we have implemented:

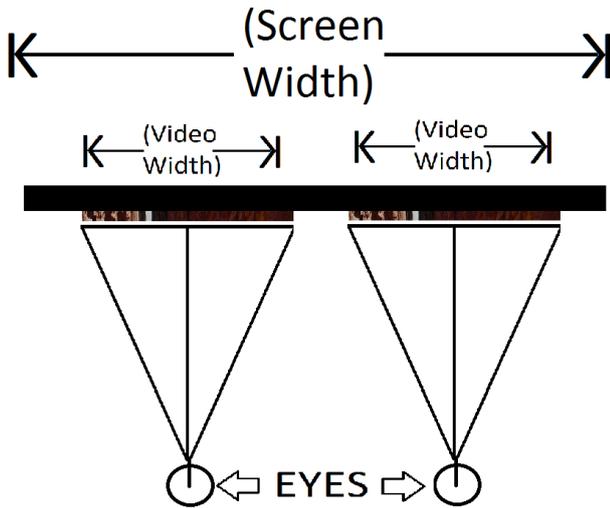
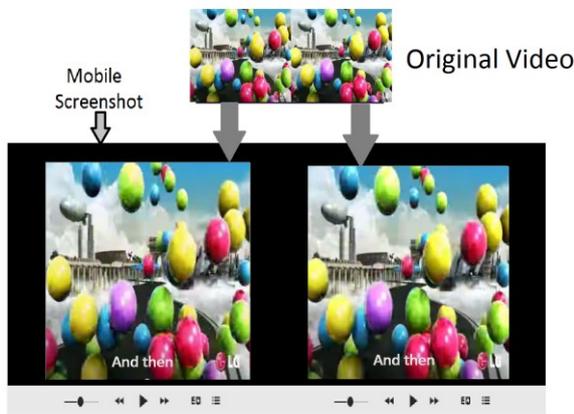


Fig. 7. Aligning of the cropped video to the centre of each "texture-video-view".



Cropped and centered video using textureVideoView's.

Fig. 8. Input video and the output video.

For the aspect ratio purpose, we have made a preset of aspect ratio. It is because, often times, due incorrect aspect ratio of specific videos, it does not let the user have a correct grasp of 3D. And, the user mostly end-up seeing two distinct images at such times. Also if none of the presets work fine, user can further manually resize the aspect ratio by selecting manual-mode and then pinch-zoom the video-view until the correct 3D image is formed.

The controls for each "texture-video-view" are independent and do not overlap in half. And, we've implemented the gesture recognition on both "texture-video-view", for better control.



Fig.9 . Various features implemented using gestures in specific areas for each view.

The correct speed of both videos is also one of the major problems i.e., when the two frames are not synchronous as shown in the fig. 5. For such times user can skip the few frames from either of the texture-video-view to adjust the speed of each video according to his needs.

For the overlapping controls and subtitles, we've aligned a surface-view above both of the controls. And the text is read from the ".srt" file and then rendered into the surface-views. This overcomes the drawback explained in the fig. 6, where the subtitles and the controls all look overlapped. The controls are shown in fig. 9 and the subtitles in fig. 8.

VI. FUTURE ENHANCEMENT

The future enhancements are mostly in the direction of making the app play the 360° videos which come in ".mp4" format. By adding the functionality of motion -sensors, user will be able to see the complete panoramic 3D video by rotating his head. Also, the navigation becomes a little bit hard, thus by adding the NFC tag features to repetitive tasks, the videos can be controlled to some extent.

VII. CONCLUSION

The 3DMP can thus prove to be a pioneering app to enhance the viewing of the user. As it helps to minimize almost all the drawbacks, that are being discussed here.

VIII. REFERENCES

- [1] Deshmukh, Kishor P., and Gauri K. Bodkhe. "3DP: The 3D Panorama."
- [2] Rhoads, Geoffrey. "Method for increasing the functionality of a media player/recorder device or an application program."
- [3] Chen, Shenchang Eric. "Quicktime VR: An image-based approach to virtual environment navigation." *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM, 1995.

[4] Mendiburu, Bernard. "3D movie making." *United States of America: Focal Express* (2009).

[5] Real-time software-based video coder for multimedia communication systems
Ho Chao Huang, Jau-Hsiung Huang, Ja-Ling Wu.

Authors:



Kishor P. Deshmukh Pursuing
B.E. (Computer Engineering)
University of Pune
Department of Computer Engineering,
Government College of Engineering and Research
Awasari (Khurd),
Taluka- Ambegaon, Dist- Pune



Gauri K. Bodkhe
B.E. (Computer Engineering)
University of Pune
Department of Computer Engineering,
Government College of Engineering and Research
Awasari (Khurd),
Taluka- Ambegaon, Dist- Pune