**Glass-free 3D Internet TV System Using Integral Imaging**

Md. AshrafulAlam1, Amit Hasan Khan1, Fairoz Nower Khan1, Noor E Nisa1, Ashraful Jannat1

1Dept. of Computer Science and Engineering, BRAC University,

66 Mohakhali, Dhaka, Bangladesh

[ashraful.alam@bracu.ac.bd](mailto:ashraful.alam@bracu.ac.bd), {[amit.hasan1010](mailto:%7bamit.hasan1010@gmail.com), [fairoznowerkhan](mailto:fairoznowerkhan@gmail.com), [nisa2912](mailto:nisa2912@gmail.com), [ajannattt](mailto:ajannattt@gmail.com)} @gmail.com

Abstract

We propose a novel technique to synchronize elemental images and audio signal and the transmission technique for a glass-free 3D TV system based on integral imaging. The main idea behind the method is to generate 3D video based on elemental images synchronized with audio stream. The system uses the depth information and RGB data of per frame of a video through Intel RealSense 3D camera and the audio stream from microphone. The audio file is sampled according to per frame duration of the video and kept in different buffers but having same index. The frames are divided into elemental images using Elemental Image Generation algorithm and the audio signal is synchronized according to the index. Then the stream of elemental images and corresponding audio data is transmitted to data server for storage. HLS streaming protocol is used to stream the TV content via. A dedicated web application was made that fetches data from the server and plays video on the user end display device. By using multi-array of lenses in front of display, the video is viewed as three-dimensional with the help of integral imaging technology. This omits the need of wearing 3D glasses.

**Keywords**:Elemental image, 3D TV, Glass-Free, Integral imaging, Streaming protocol, Internet TV, Synchronization.

# INTRODUCTION

3D television (3DTV conveys depth perception to the viewer by employing techniques such as stereoscopic display17, multi-view display16, 2D-plus-depth, or any other form of 3D display. Most modern 3D television sets use an active shutter 3D system or a polarized 3D system, and some are auto stereoscopic without the need of glasses18. There are several techniques to produce and display 3D moving pictures19, 20.

The future of 3D television is also emerging as time progresses. New technology like WindowWalls (wall-size displays) and Visible light communication are being implemented into 3D television as the demand for 3D TV increases. One might be able to obtain information directly onto their television due to new technologies like the Visible Light Communication that allows for this to happen because the LED lights transmit information by flickering at high frequencies.

The continuously increasing demand of new media technologies needs the rapid improvement in 3D display technology. In present time, 3D glasses are required to convey the depth perception of a motion picture to the viewer. These glasses only work with passive 3D technology. However, a glass free 3D technology was proposed for still images. Our proposed model synchronizes audio with per frame of motion pictures for a 3D TV system21.

The first step to attain glass-free 3D technology is integral imaging. In the past few years, Integral Imaging has garnered the attention of a good number of researchers and enthusiasts, evidenced by the surge in relevant journals and conference publications. First proposed by Gabriel Lippmann in 1908, it is a 3D imaging method that can produce full color, full parallax 3D images without the assistance of any larger objective or viewing lens. The technique is basically composed of numerous distinct elemental images viewed through an array of micro lenses. In spite of its benefits, Integral Imaging has its share of constraints including narrow viewing angle, inadequate resolution and shallow image depth. Narrow viewing angle is a major limitation which is caused by restriction in the space where each elemental image can be displayed. This drawback alone is constraining the commercialization of the technique. A number of studies have been conducted to solve it including lens switching method1, the curved lens array technique2, moving lenslet arrays with a low fill factor3, micro-convex-mirror arrays4 and multiple-axis telemetric relay system5. These methods however only can minimize the problem, but not to an extent so that it can resolve the drawbacks to the commercial implementation. A few efficient methods have been verified to solve the problem. Choi et al. demonstrated a multiple-viewing-zone integral image display using a dynamic barrier array that generates multiple viewing zones by directing the light rays emitted from the elemental images using a dynamic barrier array6.

Alam, Kwon, Piao, Kim and Kim (2015) demonstrated a viewing-angle-enhanced integral imaging display system that uses a time-multiplexed, two-directional sequential projection (TTSP) scheme and a directional elemental image generation and resizing (DEIGR) algorithm7. The system consists of three processes: acquisition of depth and color information of a real object using sensor, generation of two sets of directional elemental images contemplating two different angular perspectives using a DEIGR algorithm, and projection of two sets of DEIs using the TTSP scheme. The experiment conducted during the research demonstrated that the viewing angle can be enhanced almost two times compared to other existing methods.

Integral imaging is mainly composed of two parts: pickup and display. During pickup, the object is imaged using an array of lenses, and the image is produced as an array of parallax images of the object, termed an elemental image array (EIA). To implement a system that can be used to image real-world objects, the EIA of the objects must be generated in real time, with the system capable of supporting real-time computation of the EIA. Jeong et al. proposed a system for generating EIAs using a central processing unit (CPU) and graphics processing unit (GPU)15. This method utilizes the parallel computing architecture of the GPU, allowing concurrent computation of the EIA from the real-world objects in real time. A depth camera acquires depth and color data of real-world objects. Then GPU parallel processing is applied allowing real-time computation of the EIAs for real-world objects. The generated EIAs are displayed on an LCD panel using a lens array and the developed 3-D images are shown to the viewer. The experiment conducted during the research shows elemental images generated at a rate of more than 30 frames per second using a quad-core 3.0 GHz CPU and a 256-core GPU.

# methodology

A unique system is proposed here to synchronize elemental images and audio signal and the transmission technique for a glass-free 3D TV system based on integral imaging. The main idea behind the method is to generate 3D video based on elemental images synchronized with audio stream. The process is divided into 5 phases- acquisition, processing, synchronization, transmission and 3D reconstruction at the receiving end. Figure 1 illustrates the workflow of whole system.

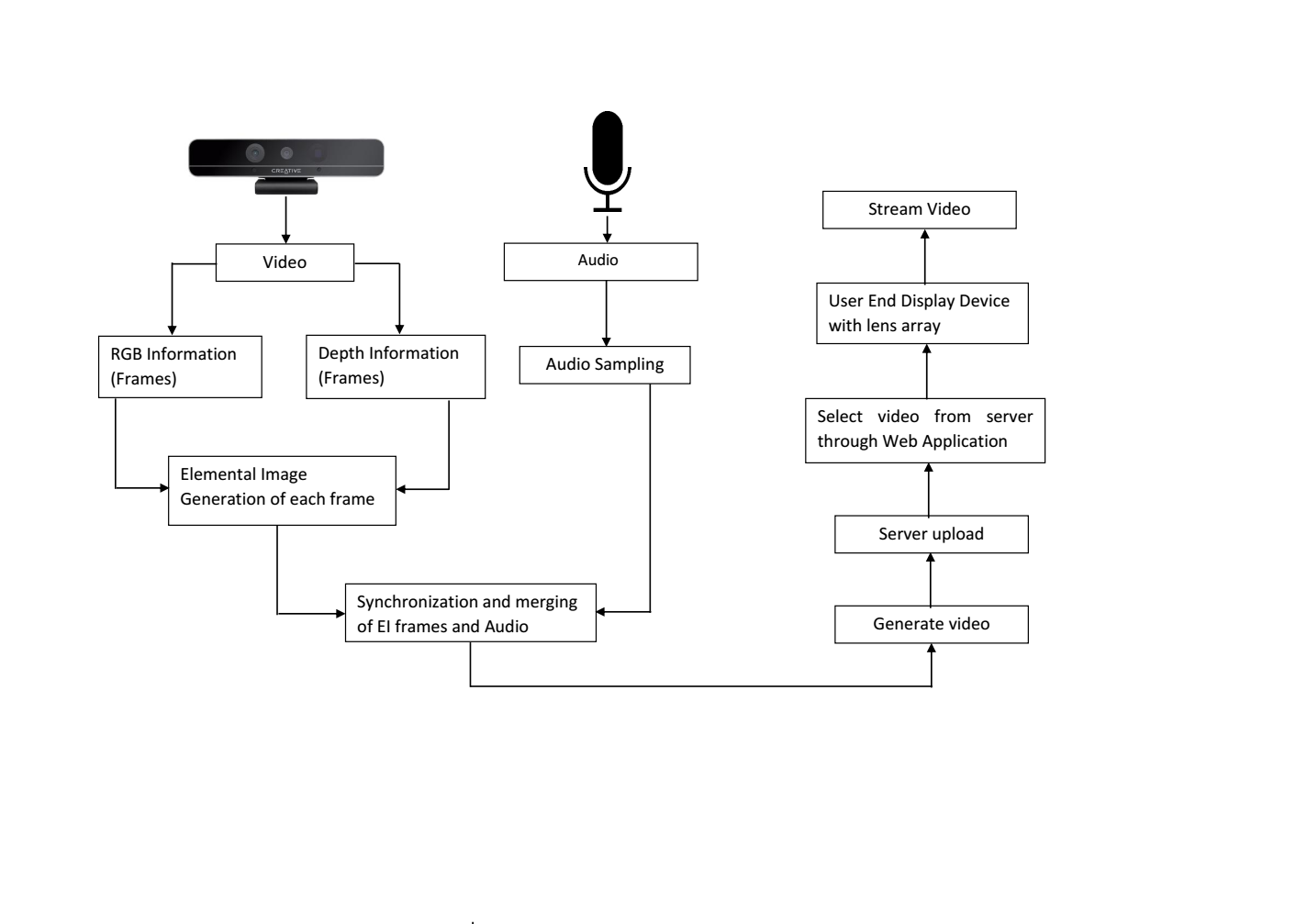


Figure 1. Block diagram representing the entire system.

## Acquisition

The acquisition phase for the proposed system requires three types of input data- Depth image, RGB color image and audio sample.

Depth image gives the “depth” of the object or the “z” information of the object in real world. The intensity values in the image represent the distance of the object from a viewpoint. It can be color coded to visually represent the close and far objects efficiently. Depth maps can be obtained using Stereo camera, Laser triangulation etc. and are exclusively used in many 3D vision algorithms.

Images with color can be represented with 24-bit RGB images. The colors in RGB images (24-bit with 8-bits for each of the red, green and blue channels) are used to show multi-channel images. The colors are designed to reflect genuine colors (i.e. the green in an RGB image reflects green color in the specimen). A RGB-D image is simply a combination of a RGB image and its corresponding depth image. One can use Kinect sensor8 or the more advanced Intel RealSense9 cameras to acquire these information.

In the experiment, the approximate depth and RGB information of per frame of the video is gathered using Intel RealSense camera and the audio signal is received through microphone. The depth value was calculated through Infrared projection and grey-scale depth camera. Figure 2 shows the acquisition of grayscale image (a), color image (b) and audio signal (c) frame by frame over time T, where T represents the total recording period.

1. (b) (c)

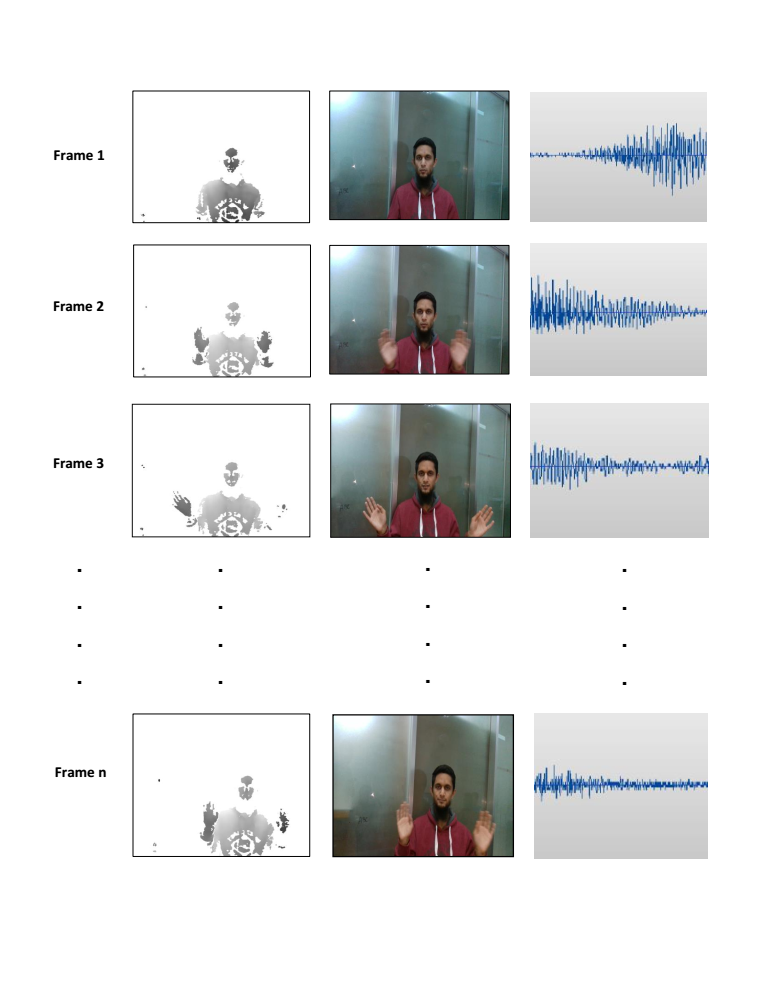


Figure 2. (a) Depth stream, (b) RGB stream and (c) audio signal recording.

## Processing

Initially the video captured from acquisition phase is extracted into frames. The Realsense SDK clip editor tool13 analyzes, visualizes and edits SDK recorded files. The tool has been used to decode video file and convert it to bitmap image frames. This created two separate streams of images- RGB color and grayscale. The audio signal is sampled using FFmpeg10, a command-line-based tool for processing of video and audio files. FFmpeg includes libavcodec, an audio/video codec library used by many commercial and free software products and supports wide variety of audio formats. The audio obtained from the microphone is sampled as per the total number of image frames. As the video is captured at 30fps, each frame consumes about 1/30 namely, 33.3333ms. Therefore, the audio is sampled with the period of 33.3333ms. Color, depth and segmented audio information are then stored in data buffers with same indexing to keep track of their sequence. It can be mapped as a matrix (1) where C, D and A denotes respectively color images, depth information and audio segments starting from index 1 to n.

(1)

The streams of RGB-D images are then processed through elemental image (E.I) generation algorithm proposed by Alam, Kwon, Piao, Kim and Kim7. During the initial stage the parameters of the lens and display components are recorded which are later used in the pixel mapping algorithm11 to generate E.I. The obtained depth data was converted as stated in the algorithm11.

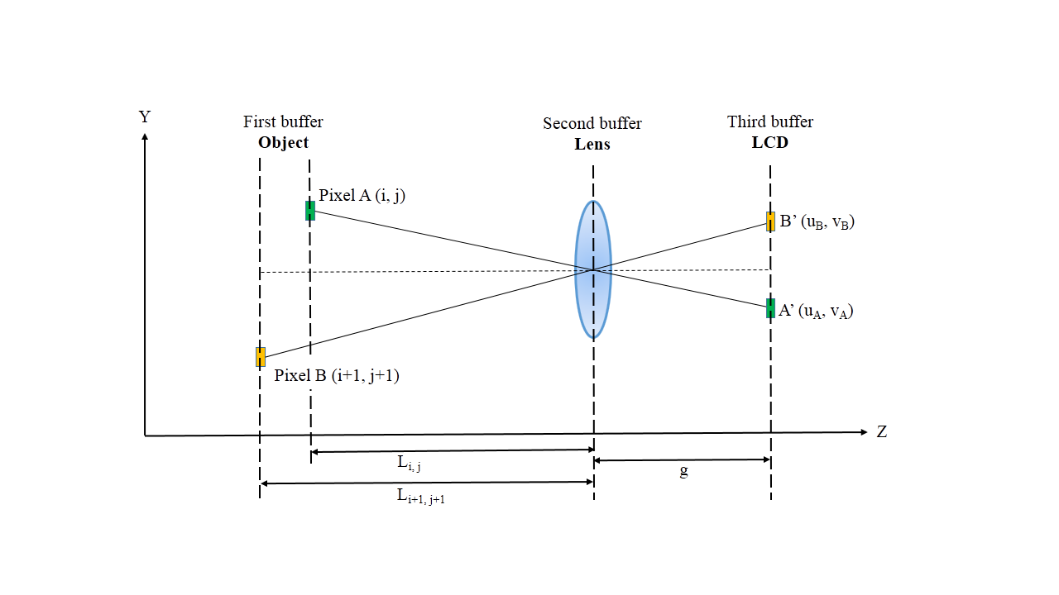


Figure 3. Illustration of pixels mapping from object image plane to elemental image plane for rendering elemental image pixels.

The central depth (d) of the system is expressed by

(2)

Where f is the focal length of the lens and g is the gap between display and lens array.

The converted distance L (i, j) between lens and object is expressed as

(3)

Where Z (i, j) is the real depth of the pixel (i, j). The L (i, j) is later used to calculate the pixel coordinates. The pixel coordinates or (u, v) is calculated by

(4)

(5)

Where i and j are the object pixel indices in x and y axes.

Implementing these formulas, the E.I generation algorithm converted each extracted frames of the video into a sequence of elemental images (example shown in figure 4).

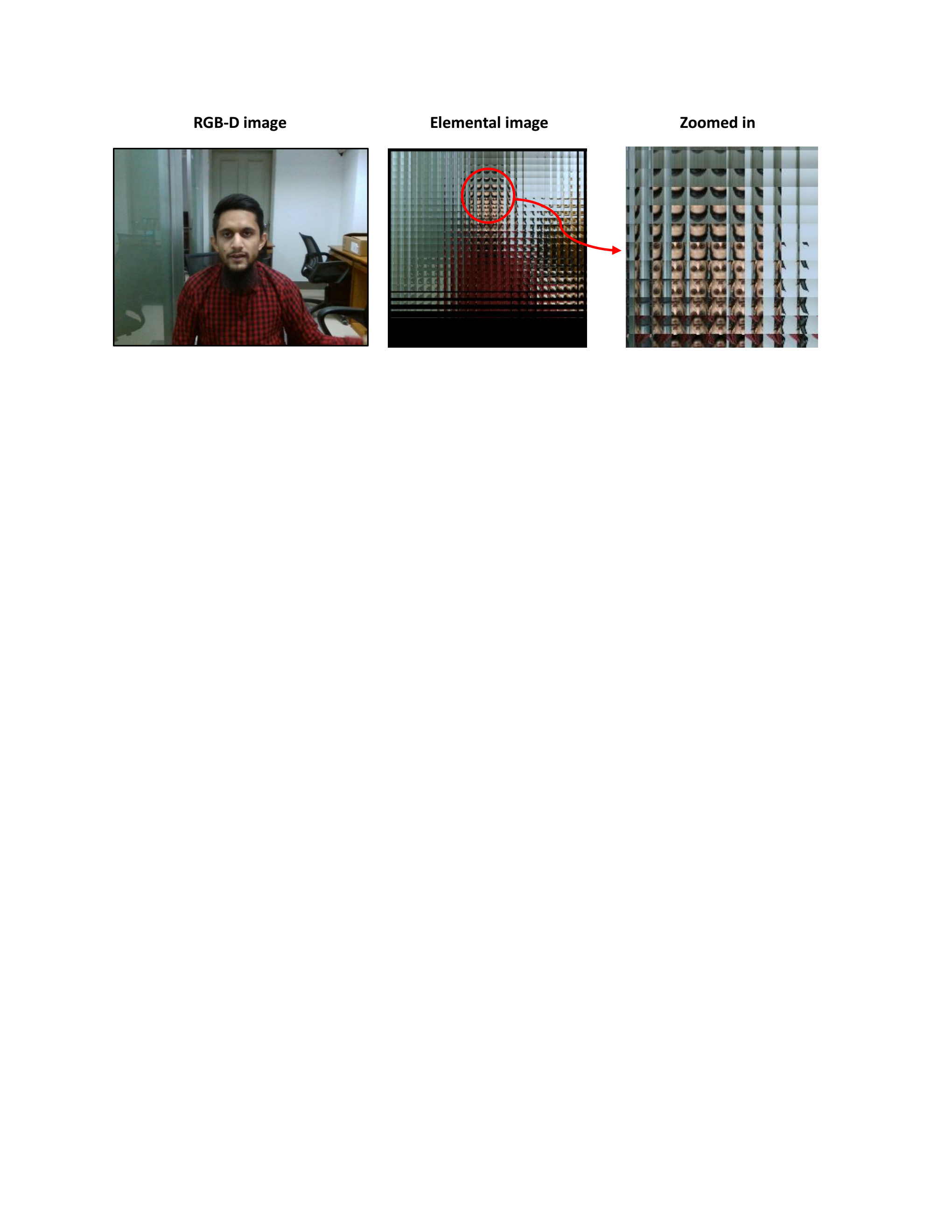


Figure 4. Generation of elemental image.

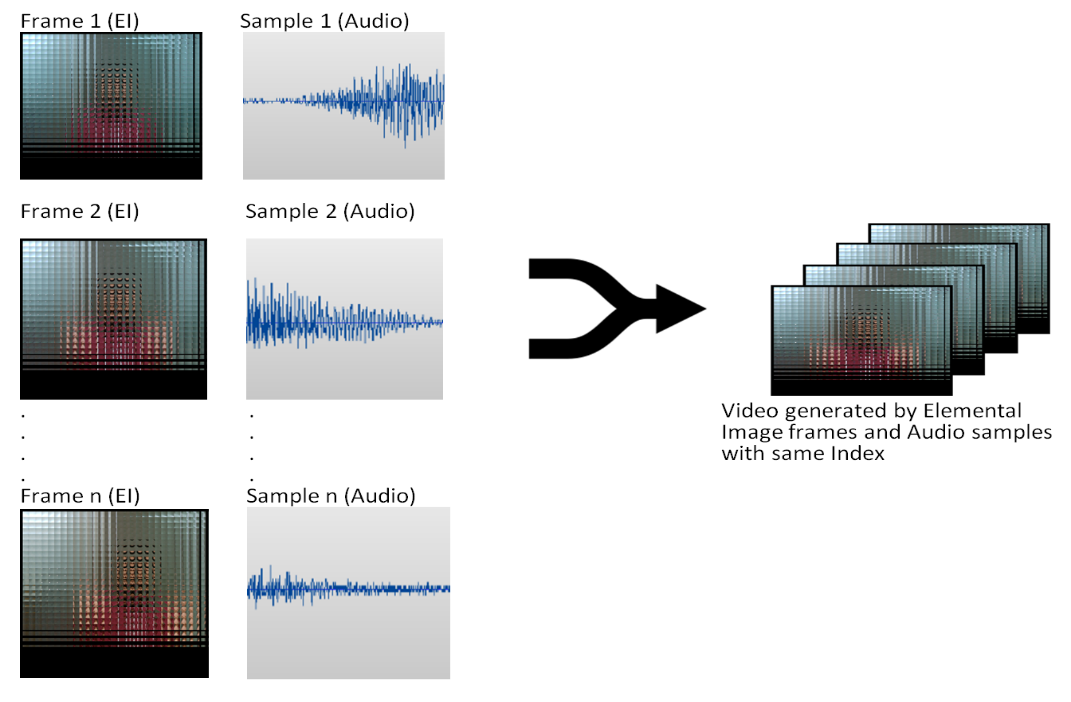
The algorithm was implemented using MATLAB and took approximately 90 seconds to generate each image. The EI are stored separately in a different buffer with its indexing kept parallel to that of the audio buffer, so that each EI is stored corresponding to its audio frame.

## Synchronization of E.I frames with Audio Samples and Video Generation

The sequence of elemental images (EI) and audio segments are stored in data buffers with identical indexing. It can now be mapped as (6).

(6)

The stored Elemental Image frames are synchronized and merged with the corresponding audio samples using FFmpeg command tool. A 30fps video file is generated after the merging of all EI frames with its respective audio samples.



FFmpeg library

Figure 5. Synchronization and merging of EI frames corresponding to audio samples.

## Transmission of 3D TV Content

Most digital video is designed for two things: storage and playback. For storage, the generated video file is uploaded to the 3D TV server (figure 6) using Transmission Control Protocol (TCP). The TCP/IP is a common protocol which provides a data transmitting mechanism to manage packet flows between devices connected to the Internet. A packet means a collection of data or the video signal in this case. The video signal is segmented into multiple IP packets so that it is ready to be sent over the IP network.

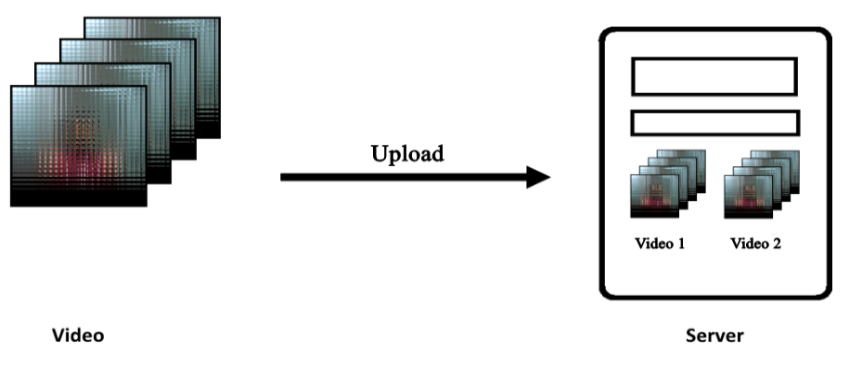


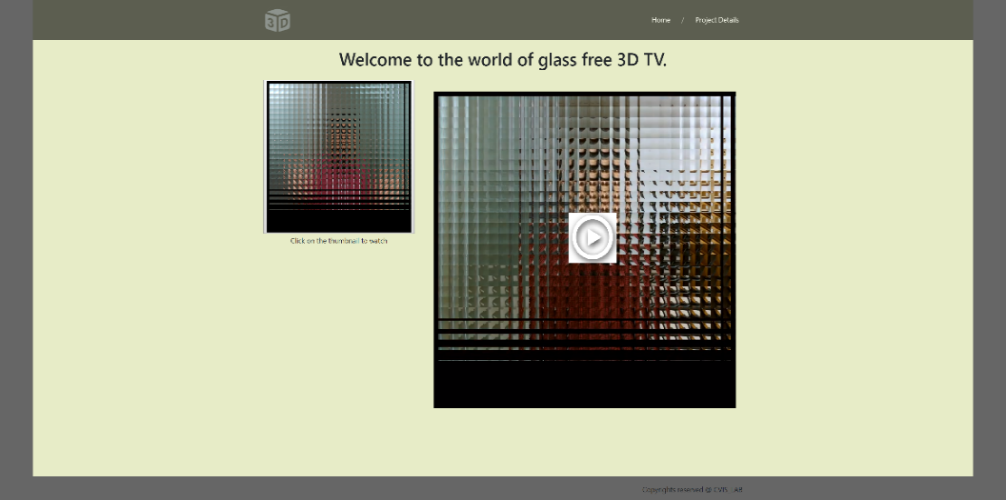
Figure 6. Transmission of TV content to server.

On the user end, the TV content can be fetched from server and streamed online at any time. Streaming a video involves breaking it into small chunks and are sent sequentially. A video streaming protocol is a standardized delivery method for breaking up video into chunks, sending it sequentially to the viewer, and reassembling it for playback. On the proposed system, HTTP Live Streaming (HLS) protocol14 has been used. The HLS protocol chops up video content into short chunks. HTTP then delivers these short clips to viewers. HLS streams are generated on the fly, and an HTTP server stores those streams. The protocol splits video files, as mentioned above, are into short segments with the .ts file extension (standing for MPEG2 Transport Stream). It adapts the video quality according to the internet speed at the user end. HLS is a widely used streaming protocol that supports desktop browsers, smart TVs, and both Android and iOS mobile devices. HTML5 video players also natively support HLS. This allows the stream to reach as many viewers as possible.

## Reconstruction of 3D TV Content at User End Application

* + 1. **Video Playback via Web Application**

A web application was built to make the system available and accessible on worldwide web. The app was created from scratch using HTML and CSS to build the structure and design the user interface. Furthermore, Bootstrap was used to make the app responsible so that it is easily viewable on any device. Moreover, JavaScript is used to make the website fully functional. Figure 7 shows the interface of end user.



Video thumbnail

Video player screen

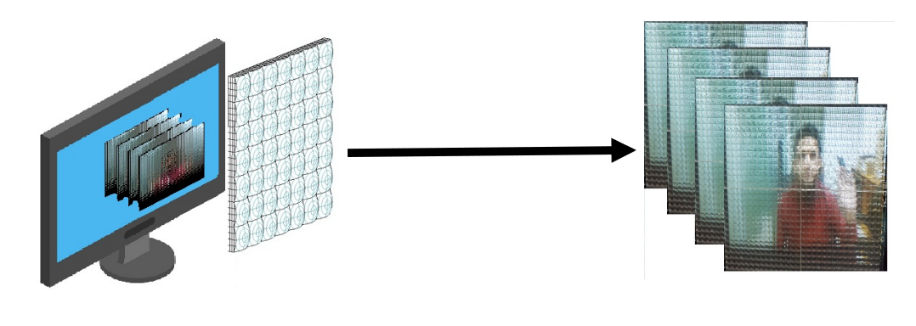
Figure 7. Snapshot of the user end web application.

The application consists of two parts. On the left side there are thumbnails of each video and on the right side one particular video is displayed by default. User can watch any of the videos by clicking on the particular thumbnails provided for each video. All the videos are pre-uploaded to the server. Domain and hosting service was bought to make the app active on the internet. There is one more page on the website where the project documentation and features are given. The app has a user friendly interface.

The selected video is fetched from the server using HTTP and HLS protocol. When the video is streamed to a viewer, it is imperative that the bits are delivered quickly so as to prevent a video from unwanted delays or buffering. The data being sent is also time-sensitive as slow data streams result in poor viewing experience.

* + 1. **Reconstruction of 3D Content using Integral Imaging**

The video is reconstructed at the display monitor through Integral Imaging. As mentioned earlier, integral imaging consists of two parts; pickup and display. During displaying, the obtained elemental images on the user end monitor, light from the display device returns through the lens array and projects the generated elemental images onto the focal plane of the lens array12. The array consists of 30 by 30 number of lens which is in accordance to the number of elemental image segments. It is placed with a distance of 15mm from the display and in a way so that it is perfectly aligned with the video. The overlap between all the projected elemental images coincides in the 3-D space to form a real 3 dimensional image, which appears reversed since the user’s perspective is opposite to that of the lens array. The depth inverted images are transformed to correct depth by rotating each image by 180° along the center. The reconstructed images differ according to perspective of the user. The resolution of each reconstructed image depends on the resolution of the camera used (Intel Realsense sr300 in this case) and the number of generated Elemental Images.



High-resolution monitor

3D TV content

Lens array

Integral imaging

Figure 8. 3D reconstruction using integral imaging.

# EXPERIMENTAL SETUP AND RESULTS

Our experimental setup consisted of Intel RealSense camera and microphone for the acquisition part of the experiment. Also, display monitor attached with lens array was used for user end reconstruction of the 3D TV content. The specifications of the equipment used in the experiment are given in Table 1 below.

Table 1. Specification of experimental setup.

|  |  |  |
| --- | --- | --- |
| **Key Components** | **Specifications** | **Characteristics** |
| Lens array | Pitch of lens | 5mm |
| Focal Length | 10mm |
| Number of lenses | 30x30 |
| Intel RealSense | Model | SR300 |
| Color Resolution | 1920x1080 pixels |
| Depth Resolution | 640x480 pixels |
| Microphone | Model | Samsung |
| Display Monitor | Model | ASUS LED monitor |

The experiment was conducted on four separate videos with durations of 3s, 5s, 7s, and 9s respectively. The most crucial part of the experiment was the synchronization of EI frames with the right audio samples. At the acquisition part shown in Figure 1, we made sure that the camera and microphone were turned on at the same time and also the duration of the audio and video files are exactly same so that the files could be equally segmented and stored with same indexing in different buffers. After the acquisition part, the Elemental Image Generation Algorithm was able to successfully process the depth and RGB values and the generated elemental images were stored with the same indexing as the depth and RGB information. Then, we could successfully merge and synchronize the audio samples with corresponding EI frames for all four videos with full accuracy. However, the only drawback was that the elemental image generation process is time consuming as for each frame it took approximately 90s to generate elemental image.

The reconstruction of the 3D TV content at the user end determines the experimental result. The multi-array lens was carefully aligned with the video appearing on the user end web app and was kept at 15mm distance from the display monitor. The video is reconstructed based on integral imaging and the user is able to watch 3D content with synchronized audio. However, the result could have been improved if we could use lens array containing smaller pitched lens. Additionally, the 3D content can be watched at the viewing angle of approximately fifteen degrees from the user end display system. Nevertheless, we got the best result for all four videos when the viewing angle is at approximately zero degree. The result at the user end shows a smooth 3D video perfectly synchronized with audio. The reconstruction for two videos out of the four are shown in Figure 9.

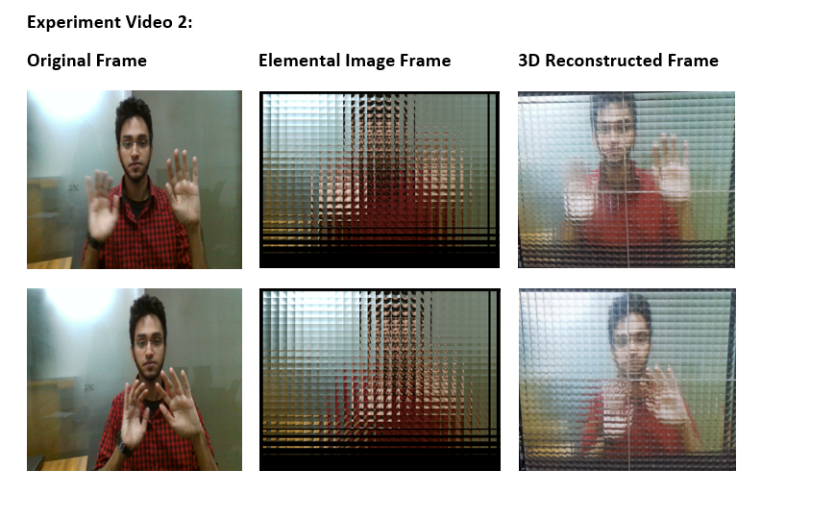
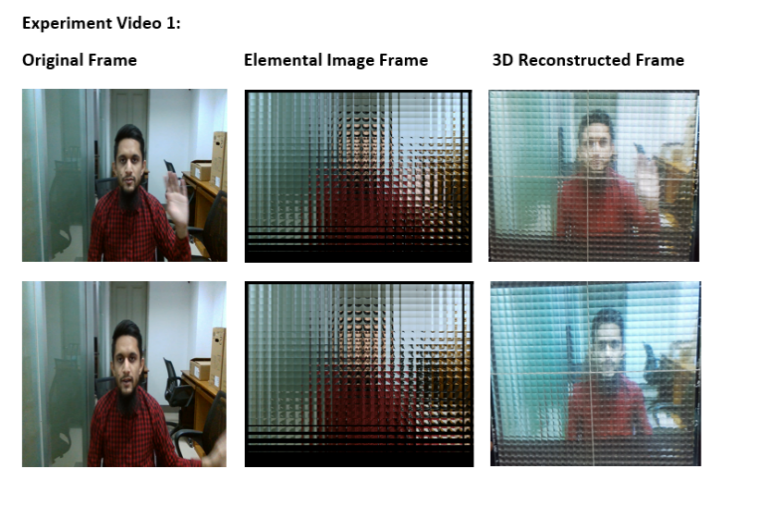


Figure 9. Video frames, respective elemental images generated through E.I generation algorithm, 3D reconstruction using integral imaging of two videos.

Additionally, samples of captured color video, synchronized elemental image video and 3D reconstruction can be found on ([media 1](https://drive.google.com/open?id=1CJizYMq0YjFnmBua7LWsqGgz8GWNbRo_), [media 2](https://drive.google.com/open?id=1iJHbcXab7yZv7jKI1kcB0167oe0zZqnG), [media 3](https://drive.google.com/open?id=18RsvmpvUU8bci66vGBsKVxVnWHcB7wuP)) respectively.

# CONCLUSION

From this study, we established a glass free 3D Internet TV system by the synchronization of Elemental image frames and audio samples converting it into a video file. The elemental image frames are generated with the depth and RGB values of the initial video. The EI video with synchronized audio is then uploaded to the 3D TV server and can be played from the user end web application. Furthermore, the user end display system consists of multi-array lens by which the video is reconstructed based on integral imaging. The user experiences 3D video with synchronized audio at the user end display.

We have implemented the elemental image generation algorithm using MATLAB which is a CPU centered task. Since MATLAB is an interpreted language, the execution speed is a major limitation. It took approximately 70 seconds to generate each elemental image of a certain frame. In future, we will integrate the system to GPU parallel processing using NVIDIA graphics unit and their video codec SDK. This will ensure faster generation of elemental images and therefore, a more clean and efficient system.

As video and audio processing occurs in a number of steps, the lack of ability to recover means that all steps must be carefully designed to avoid drift. Additionally, even though each stage causes only minor drift, the drift can still be accumulated into an obvious one. Despite careful planning and timing calibration, errors can creep in during processing stage and measures must be taken to identify the factors that contribute to any delays caused and manage the resulting misalignment. Also, the timing of turning on the camera and microphone is extremely crucial as it determines the accuracy of synchronization of the end product. Thus, the acquisition part of the experiment has to be very precise for the best outcome.

Code efficiency is a key element to ensure a high performance and effective system. The algorithms that we have implemented in different stages like the elemental image generation or the video encoding/decoding algorithm- their efficiency and runtime execution is directly linked to code efficiency. Achieving code efficiency can be done by removing redundant segments, using optimal memory and reusable components, using best keywords, data types and variables, calling function in place of repeated code etc. We have used MATLAB to implement the algorithm and the since the execution speed is a disadvantage in future we would like to use other languages like C# or C++ which would make the program faster, optimal and more efficient.

Previously, glass free 3D system could only display still images. With our algorithm, it will be able to display 3D videos. Our algorithm is developed to make the best use of latest 3D camera resources. Thus it will deliver a considerable FPS (frame per second) value and ensure pleasant watching.

References

1. Lee, B., Jung, S., Park, J., "Viewing-angle-enhanced integral imaging by lens switching", Optics Letters 27(10), 818 (2002).
2. Kim, Y., Park, J., Min, S., Jung, S., Choi, H., Lee, B., "Wide-viewing-angle integral three-dimensional imaging system by curving a screen and a lens array", Applied Optics 44(4), 546 (2005).
3. Jang, J., Javidi, B., "Improvement of viewing angle in integral imaging by use of moving lenslet arrays with low fill factor", Applied Optics 42(11), 1996 (2003).
4. Jang, J., Javidi, B., "Three-dimensional projection integral imaging using micro-convex-mirror arrays", Optics Express 12(6), 1077 (2004).
5. Martínez-Cuenca, R., Navarro, H., Saavedra, G., Javidi, B., Martínez-Corral, M., "Enhanced viewing-angle integral imaging by multiple-axis telecentric relay system", Optics Express 15(24), 16255 (2007).
6. Choi, H., Min, S., Jung, S., Park, J., Lee, B., "Multiple-viewing-zone integral imaging using a dynamic barrier array for three-dimensional displays", Optics Express 11(8), 927 (2003).
7. Alam, M., Kwon, K., Piao, Y., Kim, Y., Kim, N., "Viewing-Angle-Enhanced Integral Imaging Display System Using a Time-Multiplexed Two-Directional Sequential Projection Scheme and a DEIGR Algorithm", IEEE Photonics Journal 7(1), 1-14 (2015).
8. “Kinect for Windows”, Microsoft, <<https://developer.microsoft.com/en-us/windows/kinect>> (26 January 2019).
9. “Intel® RealSense™ Technology”, Intel, <<https://www.intel.com/content/www/us/en/architecture-and-technology/realsense-overview.html>> (26 January 2019).
10. “About FFmpeg”, FFmpeg.org, <https://www.ffmpeg.org/about.html> (26 January 2019).
11. Li, G., Kwon, K. C., Shin, G. H., Jeong, J. S., Yoo, K. H., & Kim, N, “Simplified integral imaging pickup method for real objects using a depth camera,” Journal of the Optical Society of Korea, 16(4), 381-385(2012).
12. Javidi, B., Shen, X., Markman, A., Latorre-Carmona, P., Martinez-Uso, A., Martinez Sotoca, J., Pla, F., Martinez-Corral, M., Saavedra, G. et al., "Multidimensional Optical Sensing and Imaging System (MOSIS): From Macroscales to Microscales", Proceedings of the IEEE 105(5), 850-875 (2017).
13. "Tool: Clip Editor", Software.intel.com, <https://software.intel.com/en-us/node/725601> (26 January 2019).
14. "When Should You Use HLS Streaming?”, DaCast, 2018, <https://www.dacast.com/blog/hls-streaming-protocol/> (26 January 2019).
15. Jeong, J. S., Kwon, K. C., Erdenebat, M. U., Piao, Y., Kim, N., & Yoo, K. H, “Development of a real-time integral imaging display system based on graphics processing unit parallel processing using a depth camera”, Optical Engineering, 53(1), 015103 (2014).
16. Matsushima, J., Ibata, Y., Masumura, K., Asai, T., Sato, T., Shigemura, K., “Two-way multi-view 2-D/3-D display combining LC lens and HVxDP panel using novel pixel arrangement,” Journal of the Society for Information Display 24(4), 252–261 (2016).
17. Mcintire, J. P., Havig, P. R., Geiselman, E. E., “Stereoscopic 3D displays and human performance: A comprehensive review,” Displays 35(1), 18–26 (2014).
18. Bogaert, L., Meuret, Y., Smet, H. D., Thienpont, H., “Analysis of two novel concepts for multiview three-dimensional displays using one projector,” Optical Engineering 49(12), 127401 (2010).
19. Alam, M. A., Piao, M.-L., Bang, L. T., Kim, N., “Viewing-zone control of integral imaging display using a directional projection and elemental image resizing method,” Applied Optics 52(28), 6969 (2013).
20. Alam, M. A., Baasantseren, G., Erdenebat, M.-U., Kim, N., Park, J.-H., “Resolution enhancement of integral-imaging three-dimensional display using directional elemental image projection,” Journal of the Society for Information Display 20(4), 221 (2012).
21. Kwon, K.-C., Jeong, J.-S., Erdenebat, M.-U., Lim, Y.-T., Yoo, K.-H., Kim, N., “Real-time interactive display for integral imaging microscopy,” Applied Optics 53(20), 4450 (2014).