# Light Field Frame Translation

1st Anahita Nik aien
*Computer Science*
*Memorial University of Newfoundland*
St. John's, Canada
anikaien@mun.ca

2nd Nicholas Wells
*Computer Science*
*Memorial University of Newfoundland*
St. John's, Canada
nwwells@mun.ca

3rd Matthew Hamilton
*Computer Science*
*Memorial University of Newfoundland*
St. John's, Canada
mhamilton@mun.ca

***Abstract*—In this project, we implement an algorithm for light field frame translation, a technique that focuses on translating frames within a light field to achieve optimization of depth resolution of light field images of a given 3D scene. By translating the frame of reference for a light field image of a 3D scene, the 3D scene can often be imaged at a similar level of detail but with a light field image with a much lower directional resolution. We calculated the ray direction and the perspective intersection point on the introduced frame while adjusting for the frame's changes in size, focal length, and directional resolution and preserving the image quality. The proposed method is then tested on synthetic graphics that are pre-rendered into a light field image and a simulator is used to visualize the scene. The effectiveness and accuracy of our method is analyzed by comparing the result of the frame translation to a manually created translated light field image in Maya. The suggested solution maintains the quality and realism of the translated frames while enhancing computational efficiency. This can further extend to more complex non-planar 4D representations of light fields and shifting the respective frames to achieve maximum optimization.**

***Index Terms*—Light Field, Frame Translation, Ray tracing, Visual computing**

## I. Introduction

The concept of light field was popularized by Levoy and Hanrahan [1] when they proposed line parametrization by their intersections with two planes in arbitrary positions, this laid the groundwork for many subsequent advancements in light field technology. Building on this foundation, our project focuses on designing and implementing an algorithm for light field frame translation, a process that involves shifting the frame of reference within a light field representation to modify the depth and perspective of the captured scene. This technique utilizes ray tracing to accurately recalculate the intersection points and direction vectors of light rays, thereby generating new images from the adjusted frame and enabling optimization of depth resolution.

Burnett [2] introduced a light-field rendering technique used to generate hogel data by processing all directions for a single hogel simultaneously called the double-frustum hogel-rendering algorithm. However, frustum regions often contain a significant amount of free space that cannot be efficiently removed through compression, leaving the front or back frustum empty of valuable data. Therefore, frame translation can be used to optimize rendering efficiency by concentrating resources on the relevant parts of the scene that need to be rendered, improving overall performance.

The result is tested on light field images from high-quality virtual static scenes which are generated from pre-rendered images using an advanced rendering engine, Otoy's Octane.

We use an advanced simulator developed by Wells [3], that enables dynamic generation of views from pre-rendered light field images and supports real-time ray tracing, allowing us to render scenes from multiple angles using light ray information. To evaluate the effectiveness and accuracy of our algorithm, we manually translated the original scene created in Maya and conducted a comparative analysis with our translated output. The result demonstrates that depth resolution of light field images is enhanced through ray tracing techniques and the algorithm effectively translates the light field frames and shifts the depth range of the captured images.

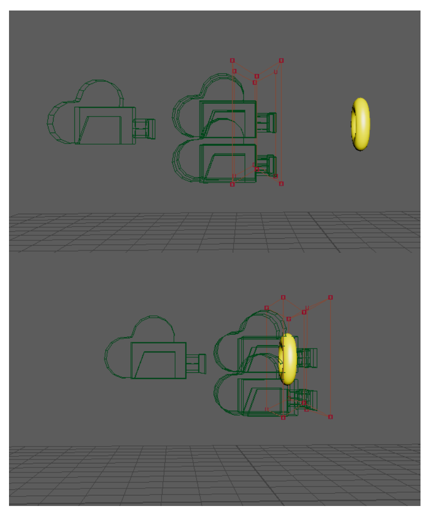

Fig. 1. Visual Representation of Light Field Frame Translation Using Maya.

## II. Related Work

The concept of light fields and rendering techniques has been covered in various research papers, significantly evolving over the years. This section touches on key research conducted on loading light fields, planar representations and their role in light field frame translation. Levoy and Hanrahan's [1] work pioneered the concept of light field rendering by introducing a robust representation of the light entering a camera (from multiple directions). Using this representation, the coordinate system on the first plane is denoted as (u, v) and on the second plane as (s, t), with light rays traced from a starting point on the first plane to a corresponding point on the second plane. Their approach treated the interaction of light with any object as a higher- dimensional matrix, effectively capturing the flow of light through unobstructed space. This was a milestone in light field research that allowed Levoy [4] to expand the theory and application of light fields, demonstrating that by capturing a large number of images from 4 various positions, it is possible to reconstruct new views of a scene. Building on their representation of the interaction of light with objects; this method, known as light field rendering, interprets the scene as a 4D array of pixels enabling the creation of accurate, perspective-shifted images by extracting the specifically oriented 2D slices from the 4D light field array, through pixel selection and interpolation. Further research into light field rendering techniques has focused on improving both efficiency, image quality and real time display of light fields.

The simulator used to test our result is developed by Wells [3]. It is based on real-time ray tracing system for light field image generation. Wells' simulator dynamically generates views from pre-rendered light field images, allowing us to efficiently render a scene from multiple angles. Using this tool has been particularly instrumental in testing our algorithm, as it provides a platform to determine the accuracy of our light field frame translation approach. This simulator was a further improvement on Hamilton et al [5] approach to creating a light field simulation based on a hybrid rasterization ray-tracing approach, which results in less ray controlled rendering than a pure ray-traced approach. In summary, prior works on light fields, such as those by Levoy, Hanrahan, and Wells, have established critical concepts and methodologies for capturing, processing, and rendering light field images. Our project builds on these advancements, focusing specifically in optimizing light field frame translation.

## III. Background

The proposed algorithm relies on the 4D representation of light field frames [1], where parallel planes are spaced 1 unit apart, and ray tracing techniques are employed to calculate the directional vectors of light rays in the scene [4]. The frame size, directional resolution, field of view (FOV), and focal length are configurable to match specific requirements. Using these parameters, the intersection point of each ray on the second plane is calculated, facilitating accurate translation of objects within the scene.

## IV. Methodology

### *A. Ray Tracing for Directional Calculations*

First, we need to load the light field image and assign all its data that will be used for ray tracing and determining the direction vectors. This involves reading the light field data from the input files, typically structured in a 4D representation of views. Each sub-aperture image will provide a different perspective, which is crucial for accurate depth and directional calculations. Once loaded, we initialize the necessary data structures to store pixel locations, light ray information, and camera parameters. These components will later be used to map each ray to its corresponding pixel on the image plane and adjust the focal plane during the ray tracing process.

Fig. 2. The synthetic scene created in Maya, used as original input for the frame translation process

Each hogel position refers to a precise location on the camera's sensor or image plane. The pixel position refers to the corresponding location in the final image visible to the user, with each hogel captures the rays of light contributing to a specific pixel. This creates the effect of viewing the scene from multiple angles [6].
To accurately calculate the rays of light associated with each hogel, we set the starting point of all rays at the origin $(0, 0, 0)$, with the center of the image plane positioned at $(0, 0, -1)$ along the z-axis. The goal is to compute the directional resolution for each pixel within a hogel, which helps determine the direction of the light rays that map to color values in the final image. The position of each pixel within the hogel can be computed as:

$$\mathrm{CurrPixel} = \mathrm{HogelLoc} \times \mathrm{PixelPerHogel} + \mathrm{CurrDirRes} \quad (1)$$

We need to scale the Pixel position in order to account for changes in perspective and focal length of each direction vector. First, the pixel positions are normalized within the hogel, meaning the pixel's location is scaled to fit within a range of -1 to 1. This normalization ensures that the pixel positions are relative to the center of the hogel:

$$\mathrm{ScaledPixelPos} = \left(\frac{\mathrm{CurrPixel}}{\mathrm{PixelPerHogel}}\right) \times 2 - 1 \quad (2)$$

With normalized pixel positions, directional vector calculations become simpler, as the pixel's location is now relative to the hogel's center. The ray tracing method in this case is based on a redefined field of view (FOV) representation, where a line and three points are used to describe the view. The focal length is normalized to 1, creating an angle equal to half of the FOV between the two sides of the triangle. This normalization allows us to calculate the maximum distance of the directional vector from the center of the hogel to the edge of Field of View using the tangent.

$$\mathrm{MaxDist} = \tan\left(\frac{\mathrm{FOV}}{2}\right) \times \mathrm{NormalizedFocalLen} \quad (3)$$

Once the maximum distance is calculated, the next step is to determine how far the scaled pixel lies from the center of the field of view to accurately trace how light rays travel and intersects in the scene.

$$\text{DistToCenter} = \text{ScaledPixelPos} \times \text{MaxDist} \tag{4}$$

This calculation is crucial for accurately tracing the rays and their intersections in the 3D scene. All these calculations must be performed for both the x-axis and y-axis to ensure accurate ray direction and perspective in the 3D world.

### *B. Frame Translation*

The goal of this step is to accurately map the light ray position that intersects with the respective plane, considering the new spatial and directional resolution, FOV and focal length adjustments. The method we use to calculate the distance from the center on the new plane is based on triangle properties and proportionality theorem. To integrate the concept of proportionality into the frame translation, we treat the two parallel lines as the two planes (representing the old and new frames), the calculated distance (from the center of the old frame), and the corresponding distance on the new frame. The two sides of the triangle are formed by the new focal length and the directional vector derived in the previous calculation. The formula to compute the new distance from the center is:

$$\text{DistToCenter} = \text{SrcDistToCenter} \times \frac{\text{NewFocalLen}}{\text{OldFocalLen}} \tag{5}$$

After calculating this distance for each pixel, it remains a normalized value (image coordinates). To transform this into real-world light field image values, we must consider the actual physical dimensions and changes in the field of view (FOV). First, we compute the new maximum distance, which represents the farthest distance a ray can travel from the center to reach the edge of the FOV in the new frame:

$$\text{NewMaxDist} = \tan\left(\frac{\text{NewFOV}}{2}\right) \times \text{NewFocalLen} \tag{6}$$

By dividing the old distance to center by new maximum distance, we can scale the distance from the original hogel's center to fit the new FOV and hogel dimensions which we can then convert this scaled value back to pixel coordinates within the hogel. This effectively transforms the old image's pixel positions so that they are correctly translated into the new image's perspective.

$$\text{ScaledInHogelPos} = \frac{\text{OldDistToCenter}}{\text{NewMaxDist}} \tag{7}$$

$$\text{InHogelPos} = \frac{(\text{ScaledInHogelPos} + 1)}{2} \times \text{NewPixelPerHogel} \tag{8}$$

To account for the physical dimension, we need to calculate each physical hogel location. The hogel pitch is important in this process because it defines the spatial resolution of the light field in physical space for both the old and new images. The hogel pitch helps translate the pixel locations from the image coordinate system (in pixels) to the real-world physical coordinates. The following ratio (Pitch) is representative of how large is each hogel is in relative to the entire physical display surface. Respectively we need to calculate these for the new frame and compute the old physical hogel location.

$$\text{OldHogelPitch} = \frac{\text{OldHogelDim}}{\text{OldPhysicalDim}} \tag{9}$$

$$\text{NewHogelPitch} = \frac{\text{NewHogelDim}}{\text{NewPhysicalDim}} \tag{10}$$

$$\text{OldPhysicalHogelLoc} = \text{OldHogelPitch} \times \text{OldHogelLoc} \tag{11}$$

Adjusting the physical location of hogels is done by adding the scaled distance from the center, which will be in terms of real-world values, which then is used to calculate the new hogel location.

$$\text{PhysicalHogelLoc} = \text{OldPhysicalHogelLoc} + \text{DistToCenter} \tag{12}$$

$$\text{NewHogelLoc} = \frac{\text{NewPhysicalHogelLoc}}{\text{NewHogelPitch}} \tag{13}$$

Finally, we calculate the pixel's new position by centering the hogel correctly and incorporating the in-hogel pixel position:

$$\begin{aligned}\text{NewCurrPixelPos} = {} & \text{NewHogelLoc} \times \text{NewPixelPerHogel} \\ & + \text{NewInHogelPixelPos}\end{aligned} \tag{14}$$

This process ensures that each pixel in the new image aligns accurately with its corresponding position from the original image, adapting to the new physical dimensions, focal length, and field of view.

### *C. Light Field Reconstruction*

After accurately mapping and shifting every pixel to its corresponding position on the new frame, the next step is to transfer the color information of each pixel and reconstruct the final image. This process ensures that the visual representation in the new frame is an accurate translation of the original light field image, with adjustments to accommodate the new dimensions and parameters. As mentioned before, we calculated the current pixel position on the old image in order to determine map the color values onto the final image. Now that the new pixel positions are determined based on the old location onto the new frame we can carry the color and brightness information from the original image to the new frame. To begin the reconstruction, we create a blank frame with the specified Width and Height, calculated by considering the rays on both edges to account for all rays coverage. This new frame acts as the canvas where the translated image will be built. The method described successfully performs the frame translation by:

1. Shifting the frame of reference from the original to the new one.
2. Maintaining the visual details, including the color

information and pixel arrangement.
3. Adjusting the depth range of the light field, ensuring that the perception of depth and the 3D structure captured in the original image is preserved in the translated version.

This process allows the light field image to be reconstructed in a new frame, accurately reflecting the changes in focal length, field of view, and hogel dimensions, while still maintaining the visual characteristics.

## V. Result

The result of translation is expected to be a shift in depth, creating a zoom-in or zoom-out effect on the object but not in the traditional sense.
The process accounts for changes in depth, perspective, directionality of light rays, and adjusts the focal length and field of view (FOV), which directly affects the perceived distance between the viewer and the objects in the scene. By changing the focal length, the light rays that hit the pixels in the image change in such a way that it mimics moving the observer's viewpoint closer to the objects in the scene. This creates a zoom effect or depth shift, as if physically moving towards the object.
The objects themselves remain stationary in the scene, but the point of view (the camera or virtual observer) shifts closer or further from them, altering how large or small the objects appear. This means that objects will appear larger (if brought closer) or smaller (if moved further away), but the overall perspective and the directional light rays remain consistent with the original viewpoint.

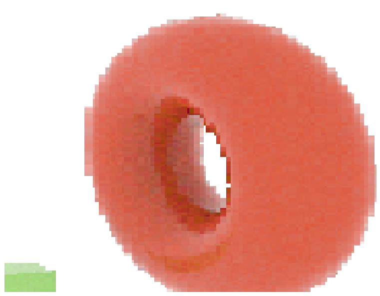

Fig. 3. Manually translated light field in Maya.

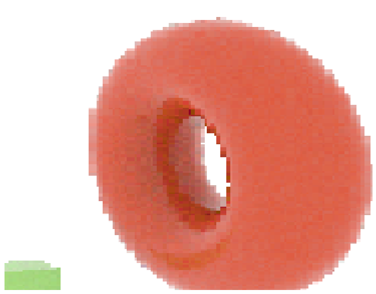

Fig. 4. Translated light field image by 1 unit.

We have synthetically created a scene in Maya, where the object is initially viewed from a specific point of view. By adjusting the position of the light field camera within the scene, we can generate a frame translation and produce a visual representation of how the manually translated result should appear.
To verify our method, we utilized the real-time light field simulator created by Wells [2] that is based on advanced ray tracing techniques, allowing digital rendering of light field images in real-time. It offers the ability to view the light field from various perspectives, making it an ideal tool for visual comparison of the original and translated frames.
The accuracy of the generated image is also evaluated by calculating the Peak Signal-to-Noise Ratio (PSNR), Mean Squared Error (MSE), and sampling frequency comparison. As shown in Figure 5, there is higher accuracy in high-frequency sampling regions, MSE of 24.54089381481 and PSNR of 34.23189984556272 indicates a moderate level of deviation between the ground truth and translated images.

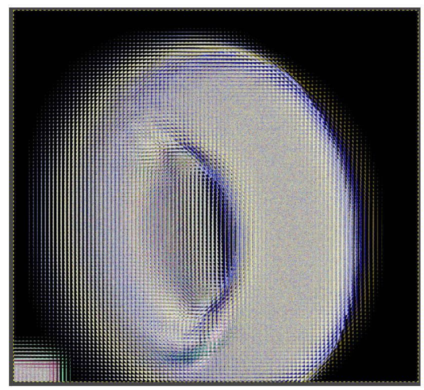

Fig. 5. Comparison of the two integral images.

We observe that the translation algorithm accurately reproduces the expected result. The objects in the scene appear closer and larger in both the simulator and Maya, demonstrating the effectiveness of our translation method. Additionally, the quality and realism of the translated frames are maintained across both platforms, showcasing the precision of our approach in handling changes in focal length, field of view, and pixel positioning. These results demonstrate the applicability of our methods for generating high-quality light field frames, which can be extended to various fields such as virtual reality (VR), augmented reality (AR), and computational photography.

TABLE I
Changes in FOV and Focal Length

| | **FOV (°)** | **Focal Length** |
|---|---|---|
| **Original Image** | 60 | 1 |
| **Translated Image** | 30 | 2 |

## VI. Conclusion

In this project, we successfully developed a method to perform frame translation of light field images with new parameters such as focal length, field of view (FOV), and hogel dimensions. Through careful pixel mapping and adjustments based on ray tracing principles, the translated frame and reconstructed light field image accurately reflects the original scene's structure at a different distance.

Using a synthetically created scene in Maya, we visually demonstrated the effect of shifting the point of view closer to the object and compared this with the output generated in a real-time light field simulator. The results from both platforms closely align, validating the accuracy of our approach. By translating the frame of reference for a light field image of a 3D scene, the 3D scene can often be imaged at a similar level of detail but with a light field image with a much lower directional resolution. This work opens possibilities for further exploration of light field image processing, including improved techniques for depth manipulation, perspective shifts, and broader applications in emerging visual technologies.

## VII. Future Work

The next step in optimizing depth resolution in light field images involves using more complex, non-planar 4D representations of light fields instead of the traditional parallel plane-based 4D representation. By shifting to non-planar surfaces, we can more accurately align with the scene's geometry and improve the efficiency of frame translation.

As previously discussed, frustum regions in parallel plane representations often contain significant amounts of free space that cannot be efficiently compressed, leaving parts of the front or back frustum devoid of valuable data. Employing non-planar 4D surfaces reduces this free space, thereby enhancing rendering efficiency and achieving greater optimization of the light field data. This approach maximizes the potential for more precise depth resolution and overall image quality.

## References


[1] Marc Levoy and Pat Hanrahan. Light field rendering. In Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques, SIG- GRAPH '96, page 31–42, New York, NY, USA, 1996. Association for Computing Machinery.

[2] Thomas Burnett. Light-field displays and extreme multiview rendering. Informa- tion Display, 33(6):6–32, 2017.

[3] Nicholas William Wells. Real-time ray traced cinematic quality light field viewer, 2021. Honours thesis, Memorial University of Newfoundland, Department of Com- puter Science.

[4] Marc Levoy. Light fields and computational imaging. Computer, 39, 2006.

[5] Matthew Hamilton, Chuck Rumbolt, Thomas Butyn, Donovan Benoit, Robert Lockyer, and Matthew Troke. Light field display simulator for experience and quality evaluation. In Digest of Technical Papers - SID International Symposium, volume 49, 2018.

[6] Yan, X., Wang, C., Liu, Y., Wang, X., Liu, X., Jing, T., Chen, S., Li, P., Jiang, X. (2021). Implementation of the real–virtual 3D scene-fused full-parallax holographic stereogram. Optics Express, 29(16). https://doi.org/10.1364/oe.434269