# Scene reconstruction using densely sampled light fields

Linda Banh, Warren Cheng, Fang-Yu Lin

## Introduction/Motivation

Scene reconstruction from image data has many wide ranging applications. It's currently used in movie production as a means of creating accurate models of movie sets and objects for post production tasks. Being able to create accurate 3D representations of a scene also apply heavily to augmented reality (AR) applications. In order to superimpose digital content onto the real world, there must be a way to map the contours of the real world so that a digital content can accurately interact with the real world. For example, a faithful 3D mesh of a room would allow a digital ball to bounce off of a real wall or coffee table rather than going right through them. Current approaches use specialized time of flight (ToF), or depth, cameras in order to accomplish this. Basically, this works with the camera emitting infrared light at different points in a scene simultaneously and using the disparity in the time of the reflected light off of surfaces in order to determine the appropriate depths of various objects. This approach works well for coarse representations of the world, but can be problematic for features that fall below the resolution of the depth camera. This is partially due in part to the fact that the patterns emitted by ToF cameras have poor spatial resolution at farther distances. Not to mention, it becomes very difficult to map the features of rooms and environments far away since depth cameras have range limitations due to the scattering of the reflected infrared light at greater distances. This makes meshing the outdoors difficult. Poor registration of the world can result in unsightly artifacts where digital content isn't perfectly occluded by real objects and can ruin the illusion of AR (See Fig. 1).

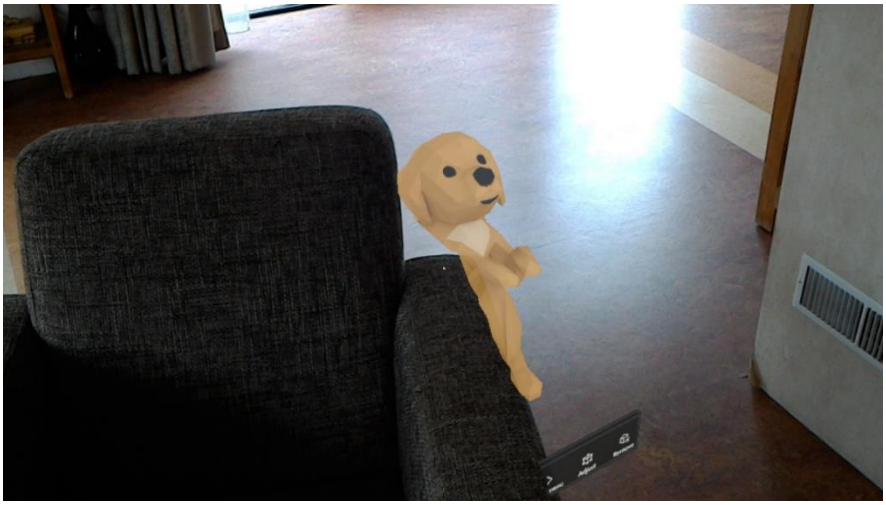

Fig. 1 - Microsoft Hololens Partially Occluded Dog [1]. Note how the dog is missing its neck and part of its body overlaps with the sofa indicative of a poor 3D mesh.

We propose a technique of using densely sampled light fields in order to produce accurate high spatial resolution depth mapped representations of scenes. We intend on using epipolar plane image data in order to extract depth data in order to reconstruct depth maps to be used in the scene reconstruction. There has been research into these and related techniques and they've garnered good results with high resolution 3D meshes and 3D reconstructions.

Ultimately, the success of this approach could bring improved 3D maps of the world for AR and bring AR realism to another level.

## Related Work

There has been a bit of research on these techniques of extracting depth information using light fields. Kim et al who worked on scene reconstruction using a set of light field images captured in a linear path [2]. They essentially generated the epipolar plane image from the light field images to generate accurate depth maps (See Fig. 2) of the scene and background. They were able to extract depth information for every scene point in their input images for both indoor and outdoor scenes allowing them to capture details at a fairly high resolution. One of the drawbacks of their approach however was that although their reconstructions were high resolution, they concluded that the absolute depth measurements had lower accuracy than compared to a laser scanner/ToF sensor approach.

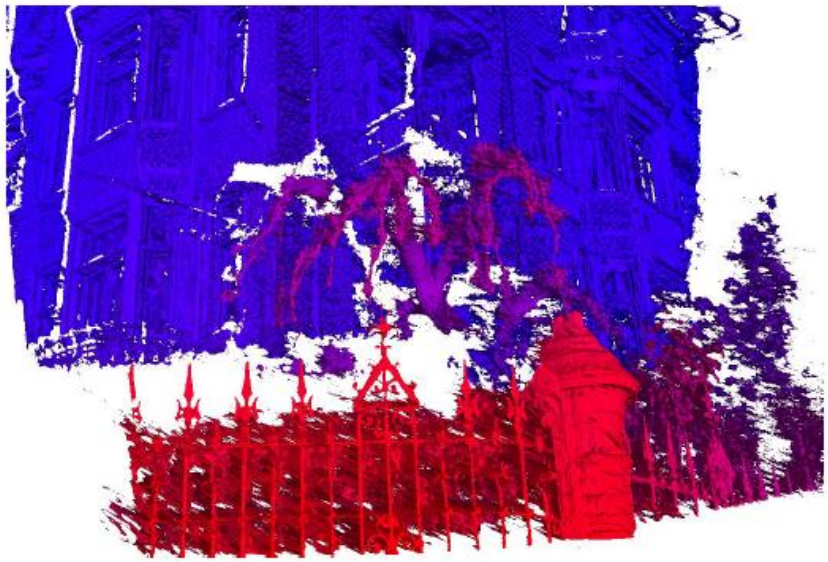

Figure 2 - 3D mesh of an outdoor environment generated from triangulating individual depth maps by Kim et. al.

Yucer and Sorkine-Hornung built on Kim et al's work by exploring a technique of using handheld video data to extract unstructured light fields (light field images captured with an unknown camera path) in order to create accurate 3D model of objects and segment them from their cluttered backgrounds [3]. They were able to extract and generate high fidelity models without knowing the motion of video camera nor the background a priori.

# Methods

Our approach for scene reconstruction is broken into 5 steps:

1. Depth Estimation
2. Edge Confidence
3. Depth Computation
4. Depth Propagation
5. Fine-to-Course Refinement

Figure 3 - Extracting epipolar plane lines from light field captures of an outdoor scene (Left). Generated depth maps (Right).

## Depth Estimation and Edge Confidence

In this project, we will take a 2D slice from a 3D image to create our epipolar-plane image (EPI). In this algorithm, however, we will use a sparse representation of the EPI, where we crop it and only use a portion of it to build our depth map. Generally, there is a lot of redundant data an EPI, and thus, this will reduce computation time (but this will be a trade-off for the variation in light field).

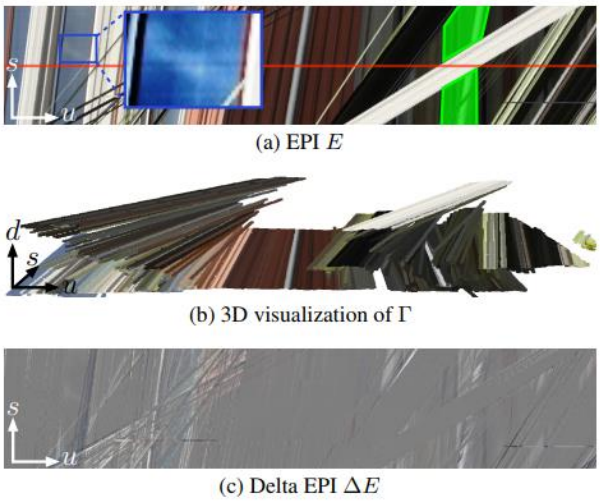


Figure 4 - Visualization of EPI E, Γ, and ΔE

We sample and store a set Γ of line segments originating at various locations in the input EPI E, until the whole EPI is completely represented and redundancy is removed as much as possible. Secondly, we store a difference EPI ΔE that will account for our reconstructed Eb and the input E. The slope (m) of a line segment associated with a scene point at distance z is represented by,

$$m = \frac{1}{d} = \frac{z}{f\,b},$$

where d is the displacement between two adjacent horizontal lines in an EPI, f is the camera focal length in pixels and b is the metric distance between each adjacent pair of imaging positions.

Using this slope,a reconstructed EPI Eb can be generated by rendering the line segments in the order of decreasing slopes (scene points from back to front).

We will then use an edge confidence to estimate how confident we feel about our depth estimation for specific parts of the EPI. This is represented by

$$C_e(u,s) = \sum_{u' \in \mathcal{N}(u,s)} \|E(u,s) - E(u',s)\|^2,$$

where N(u, s) is a 9x9 window around pixels (u, s). Ce is thresholded by 0.02 and we will use that as a binary mask. This mask will prevent us from calculating depth estimates for ambiguous pixels and hopefully improve computation time.

## Depth Computation

We will now need to assign depth z to each EPI-pixel. First, we will sample radiance values for each EPI-pixel (u, sˆ) for some disparity d,

$$\mathcal{R}(u,d) = \{E(u+(\hat{s}-s)d,\ s) \mid s = 1, ..., n\}$$

Where each n represents the number of light views. These radiance values will be used to to calculate the initial depth score,

$$S(u,d) = \frac{1}{|\mathcal{R}(u,d)|} \sum_{\mathbf{r} \in \mathcal{R}(u,d)} K\left(\mathbf{r} - \bar{\mathbf{r}}\right)$$

where r = E(u, sˆ) and K(x) = 1 − ||x/0.02||^2 if ||x/0.02|| ≤ 1 and 0 otherwise.

Our final depth estimation will be with respect to the d that maximizes S(u, d),

$$D(u,\hat{s}) = \arg\max_d S(u,d).$$

## Depth Propagation and Fine-to-Coarse Refinement

With the depth estimate found, we will propagate this value along the slope of its corresponding EPI line segment.

Lastly, we will use fine-to-coarse refinement where we will implement our pipeline on the highest available resolution so we can get all the details of the EPI (and will later use these estimates as depth bounds for coarser implementations). Next, we will apply it to coarser and coarser resolutions by downsampling the EPI.

# Milestones and Timeline

## Week 1

The focus of the first week is to acquire a dataset or render our own light field dataset of a given scene. In tandem, we intend on continuing to read up on and understand the algorithm we're planning to implement to generate the depth maps from the light field data. We will first focus on depth estimation, edge confidence, and depth computation.

## Week 2

We will begin implementing depth estimation, edge confidence, and depth computation first. We intend to continue to build out the light field processing pipeline as we believe it will take a week or so to fully bring up.

## Week 3

We will build the depth propagation and fine-to-coarse adjustment portion of the pipeline. Afterwards, we will test our processing pipeline on a simple dataset such as one from Professor Wetzstein's light field archives.

## Week 4

Our plan for the fourth week is to debug any remaining issues and to work on the final paper and/or presentation.

# References

[1] Vroegop, Dennis. *Microsoft HoloLens Developer's Guide: Create Stunning and Highly Immersive 3D Apps for Your Microsoft HoloLens*. Packt Publishing Ltd., 2017.

[2]Kim, Changil, Henning Zimmer, Yael Pritch, Alexander Sorkine-Hornung, and Markus Gross. 2013. "Scene Reconstruction from High Spatio-Angular Resolution Light Fields." *ACM Transactions on Graphics* 32 (4): 1. doi:10.1145/2461912.2461926.

[3] Yücer, Kaan, Alexander Sorkine-Hornung, Oliver Wang, and Olga Sorkine-Hornung. 2016. "Efficient 3D Object Segmentation from Densely Sampled Light Fields with Applications to 3D Reconstruction." *ACM Transactions on Graphics* 35 (3): 1–15. doi:10.1145/2876504.