Title: DiffuserCam: lensless single-exposure 3D imaging using Custom Hardware

Abstract:

In the paper by Antipa et al. [1], a lens less diffuser cam for 3-D imaging is introduced. Using a unique pseudorandom pattern of caustics on the sensor, simple calibration, and computational processing we solve for 3-D voxels with a single exposure. The use of compressive sensing and a numerical method allows us to solve the matrix inversion problem without dealing with the large original matrix. The simplified convolution forward model, validated experimentally, has a FoV of +/- 42 degrees in the x -axis and +/- 30.5 degrees in y-axis and is object dependent for resolution as reported in [2]( Supplementary Antipa). We are constructing a custom enclosure for our CMOS sensor that includes a diffuser and aperture to duplicate the results in [1]. We also followed the instructions in a tutorial reference by [1], where we used commodity parts and homemade optics where we were not able to duplicate experimental results.

1. Background

The main motivation for a DiffuserCam is to provide a light field representation of 4-D information into a 2-D sensor. While capturing spatial and angle information we want the system to be inexpensive, flexible, and compact. Providing such a system as described above falls in the realm of light-field imaging which lends itself to many applications: 3D neural activity. ( ref:Pegard et. Al), compressive radar imaging(ref:Richard Baraniuk), synthetic aperture( ref from Cai paper ref 9), and visual odometry(ref in Cai ref 11)

Why diffusers for lensless systems? At first glance a diffuser is a highly diffractive medium that would not seem to lend itself to light field imaging. A coded aperture light field is an alternative to a diffuser that can also achieve higher resolution than conventional cameras(ref: JIn paper ref 2). However, an amplitude mask system by its’ very nature limits the amount of light that the sensor can record. A micro lens array can gather more information that just x-y coordinates but suffers from scaling.

One of the key theoretical pillars in being able to use diffusers in a lensless system is that the surface of the diffuser can be modeled as a smooth Gaussian smooth surface. So the diffractive effect causing speckles (caustic patterns) from interference can be used for reconstruction. This is because each area of the diffuser is unique creating a type of signature that can encode pleoptic information about our illuminated object. These properties, of uniqueness, in theory allow us a better scaling that can be utilized by a micro lens array system. Another important point is that diffusers which are essentially phase shifter elements can concentrate the light from a 4-D light field into a 2-D sensor better than an amplitude mask using Fourier optics.

For the reasons above, a diffuser allows an encoding of the light field beyond just x-y coordinates and because of then random surface of a diffuser modeling the recorded light allows us linearity in reconstruction that lends itself to well-established inverse problem definition and optimization techniques.

<Somewhere here show the generic random matrix inversion problem. Eq 1>

< Show Figure 1 >

1. Methods

In theory a lens less system has all the information available at the sensor. However, the problem is ill-posed. Introducing a diffuser gives structure to our point spread function, or random matrix, allowing us to solve an otherwise intractable problem. If we enforce sparsity as a prior, and non-negativity (no negative pixels), minimize for the least squares we are trying to solve the equation below

[[Insert Equation (2) from Paper]]

Here the regularization term converts our image, v, into s sparse vector with the map [ use sparsity abbrv.]

< math equations >

<The problem can be sped up by the use of FISTA…>

<An even faster way to solve the inverse problem is Mention ADMM.

The math for ADMM is more involved >

For our experiments we used a Raspberry Pi Camera V2 sensor with some modifications (see discussion section). The sensor itself is a CMOS device with 3280(H) x 2464(V) active pixel count. The platform system is a Raspberry Pi 4 Model B with HDMI ports and an interface for an external sensor. For our light source we used a flashlight with a pinhole covering the LED and black adhesive to limit output other than a pin hole to simulate our point source for calibration. Another flashlight was used to properly illuminate our object of interest.

The diffusers we used were single and double-sided ordinary scotch tape. We also constructed with two paper clips that supported a mask that fit over the CMOS sensor assembly. The entire assembly was outfitted with black opaque tape to block and limit stray light other than from our point source. All the experiments were done in as dark an environment as possible. Black tape was also used for an aperture.

We did not perform a proper calibration (see discussion for more explanation). We estimated the focal distance for our camera to just a few millimeters.

1. Results

All results are using the FISTA algorithm that was outlined in the previous method section. The average computation time for reconstruction is about 30 seconds for 100 iterations. This is run on a Dell workstation that uses an Intel Core i7-8700 CPU @ 3.2Ghz with 32.0GB of RAM using Windows 10 Enterprise operating system. All code is run on PyCharm 2020.1.1 Professional Edition. Python code is downloaded and slightly modified to run on our workstation environment from [ List ANtipa reference]

Here are the results from our experiments.

< Show our PSF>

Here are results from the publication.

< Show PSF>

1. Discussion

In the original paper [1], there is a variation where the equipment used is commodity. The construction of a system while inexpensive and convenient is not practical and stable to construct consistently with success. The author followed the instructions outlined in a tutorial and had difficulty recreating the caustic pattern point spread function.

One example of a stability and repeatable solution is the conversion of a raspberry sensor (list the type that is used) for measurements. The first problem with using a sensor like the raspberry pi sensor is that the sensor is not a flat sensor and requires that the attached lens be removed to place the diffuser a few millimeters away to produce a goof caustic pattern. However, this requires some disassembly. Another problem with the use of the sensor is that the attached sensor does not lie flat as is a bit cantilevered above the printed circuit board.

A difficult problem is how to determine calibration. It is challenging to hold the diffuser so close, a few millimeters away from the CMOS sensor of the raspberry pi

A final challenge is how to with a non-varying light source, calibrate the PSF at a distance z from then sensor.

As a result, with a small investment, several hundred dollars, we propose a more stable and practical solution that increases the prospect of repeated success. As a first step we construct a custom housing using a 3-d printer. Below is a diagram of the housing. (See diagram). The housing is built on top of a sensor (list part #)

< talk about calibration and casutics> < Figure 2> Also of limitation of off-axis in (PSF model by JIn)

< Note that in our presentation of the results from Anitpa that there are many calibrations done from many depths? Although in the tutorial only a single calibration is advertised it is the authors assessment that the results published in [1] are not from a single calibration.

When trying to solve inverse problems with less than full rank, it is possible to recover the original vector if some conditions on a matrix are met. Such properties if satisfied such as Full Spark of matrix, and Null Space Property, both have full rank submatrices. A less computationally intensive way to calculate the sparsity that can be recovered is by computing the mutual coherence of the matrix, the combination of the measurement and representation matrix.

One of the major limitations that can be concluded from studying literature on lens less diffuser imaging stems from object complexity.< from Antipa and Cai> First from [1], the figure shows < the condition number>. Similar results of complexity performance are given by < ref JIn>m here the light field transmission matrix is computed by using Singular Value Decomposition.

<Use of Cai to get rid of assumptions in Antipa>

Another concern with diffuser imaging is

1. Conclusion

<The results that we obtained from our home…>

<We are currently setting up a lab to replicate the work of Antipa…..>

< Here are some specs from the system that we are trying to build >

<Also we hope to explore and model PSF with a formula similar toe JIn and also look at off -axis performance >

The paper from( reference JIN> outlines a method that can help with avoiding the use of extensive calibration in order to provide better reconstruction. The idea is based on using Fourier optics first to express the system PSF as a function with only one unknown variable, which is the diffuser phase distribution. Then JIn et al. proceeds to estimate the diffuser phase by using laser beam shaping theory. Using an architecture with a fourier lens between the diffuser and the sensor, and that fact that a far-field distribution of laser beams is proportional to the Fourier Transform of the near-field distribution, the phase of the diffuser corresponds to the phase of the diffuser which then corresponds ot the near-field component, retrievable by phase retrieval methods( here we make reference to JIn’s paper and use ref 20-22}