

# How to Build a (Pi) DiffuserCam

Camille Biscarrat and Shreyas Parthasarathy  
Advisors: Nick Antipa, Grace Kuo, Laura Waller

December 12, 2018

## 1 Introduction

In this document, you will find a general guide and introduction to building a DiffuserCam. In short, a diffuser with an aperture is placed in front of a sensor. The aperture is an opaque material that blocks stray light from entering the system. The diffuser is a transparent material that scatters light, but does not absorb it. Because of the scattering, the sensor reading is not a direct representation of the scene. The algorithm requires a single calibration measurement of how the diffuser scatters, called the point spread function (PSF). DiffuserCam's reconstruction algorithm uses this measurement to recover the original scene from the raw sensor data.

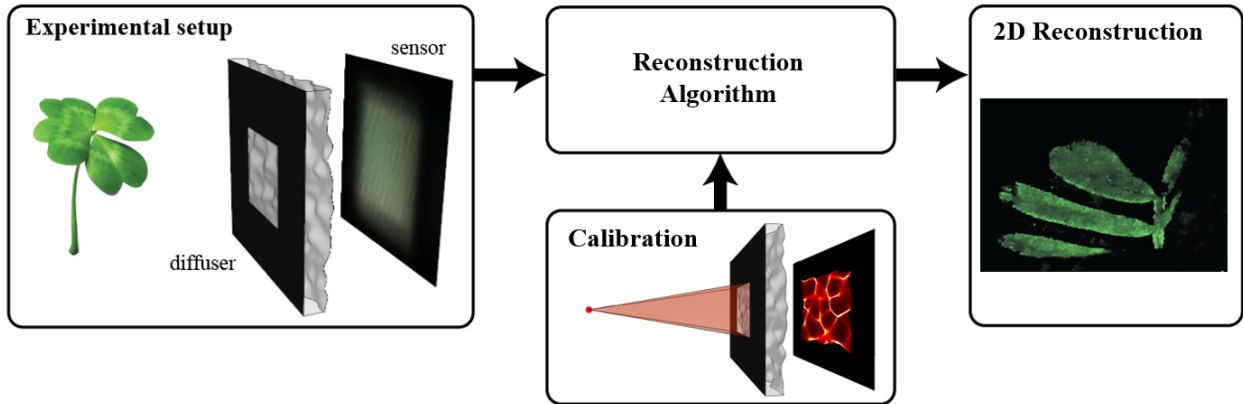


Figure 1: Setup and Reconstruction Pipeline: Only a single calibration measurement is needed. The algorithm uses this calibration measurement to take in sensor readings and output reconstructions.

## 2 Building overview

Every DiffuserCam looks like this:

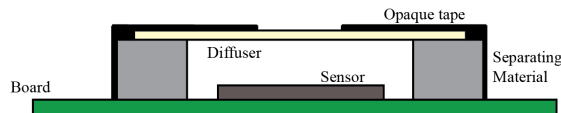


Figure 2: Side view of DiffuserCam

To build the setup, take measurements, and run the reconstruction algorithm, you will need:

1. Digital sensor: An easy way to get one is to cut open a cheap camera and remove all the optical elements until you have just the sensor. We used a Raspberry PiCamera, see Section 6 for further instructions. Note that our algorithm depends on a sensor that responds linearly to incident light; most sensors are sufficiently linear, but it's something to keep in mind. In addition, you will need to view the raw sensor reading live in order to build the camera.
2. Diffuser: Any transparent, thin, smooth material should work. In practice, we found double-sided scotch tape performs pretty well. See Section 5 for factors to consider.
3. Opaque separating material: Blocks stray light and keeps the diffuser at the right distance (black foamboard, cardboard, paper, etc.). See Section 3.2 for factors to consider.
4. Black tape: Necessary for creating the opaque aperture and blocking stray light where needed.
5. Point source: We used a single LED placed behind a pinhole aperture (see Section 4.1).
6. Computer equipped with Python 3: Our installation instructions assume you have the Anaconda distribution, but that is not strictly necessary to run the code. We suggest a computer with 8GB of RAM, but the algorithm can be run with a downsampling factor of 4 on a 4GB RAM PC with no problems.
7. Darkness: For best results, take all the pictures and calibration measurements in as dark an environment as possible (take measurements in a dark room or cover the setup with a box, etc.). Although the room should be dark, the object should be bright and well illuminated.

To build and use DiffuserCam, you will need to:

1. Find the “focal distance” of the diffuser (see Section 3.1).
2. Place the separating material around the sensor and tape the diffuser onto it so that it is propped up at the right focal distance (see Section 3.2).
3. With the liveview of your sensor open, use the black tape to create an aperture (see Section 3.3)
4. Take a calibration image (PSF) of a point source (see Section 4.1)
5. Take an image of your object, well illuminated (see Section 4.2)
6. Run the reconstruction algorithm with your point spread function and object images as inputs (see Section 4.3).

*Note: All of our example measurements are contrast-boosted for rendering in the PDF! Your measurements may have less contrast (the black is usually not so dark)*

## 3 Putting together the camera

### 3.1 Find the focal distance

The most important thing in building the camera is finding the “focal” distance of the diffuser, the distance at which the PSF has the sharpest features. At the correct distance, the features on the diffuser concentrate light, much like the bottom of a swimming pool on a sunny day (see Fig. 3). The PSF should have a similar-looking “caustic” pattern (see Fig. 4a for an example). Our double-sided tape had the sharpest caustic when placed around 3mm from the sensor. See Section 5 for information about how focal distance and choice of diffuser affect imaging conditions.

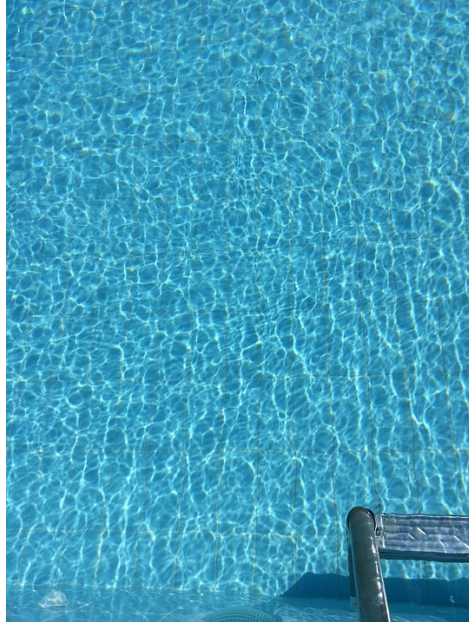


Figure 3: The surface of a swimming pool refracts sunlight to produce a very similar caustic pattern on the bottom of the pool, much like the one our diffuser creates on the sensor ([Source](#))

You should be able to find the distance,  $d$ , at which there is a clear caustic pattern by shining a point source at the diffuser in a dark room and moving the diffuser closer or farther from the sensor by hand. The point source should be “at infinity” – around 40 cm is enough for typical focal lengths.

To quantitatively find the focal distance, you can look at the autocorrelation of the PSF measurement at different depths – finding the one with the “sharpest” peak (smallest FWHM, for example) can help you quantitatively find the caustic plane (see Fig 4). That said, aligning it by eye works well enough.

### 3.2 Place the separating material

Once you’ve found the appropriate distance  $d$ , place the separating material of thickness  $d$  around the sensor, so the diffuser is propped up at that focal distance (see Figure 2). Cut a piece of diffuser that’s larger than your sensor, so it’s easier to tape down the edges onto the separator. Make sure the separating layer doesn’t let in light through the sides – test this by shining your light at the setup from different angles to see if stray light hits the sensor. We suggest connecting the separator to the sensor and diffuser to the separator with double sided tape. It is strong enough to hold everything in place, but can still be taken apart.

### 3.3 Construct the aperture

The next important thing is to place an aperture on the diffuser. Without this aperture, we wouldn’t be able to calibrate using a single image, because movement of the point source would allow new caustics to come into view on the sensor. Aside from ensuring single-shot calibration, this frame is also useful for getting a good background measurement if your sensor has a high background.

Make the aperture while viewing the sensor reading live, so you can make sure that the entire cropped PSF is within the sensor field of view (you want a black frame around the caustic reading, as in Fig 5). **Make sure the pattern occupies most of the field of view!** See Section 5.3 for a comparison. Again, test whether the aperture lets light through by shining a light source at it; add more layers of tape if it isn’t completely opaque.

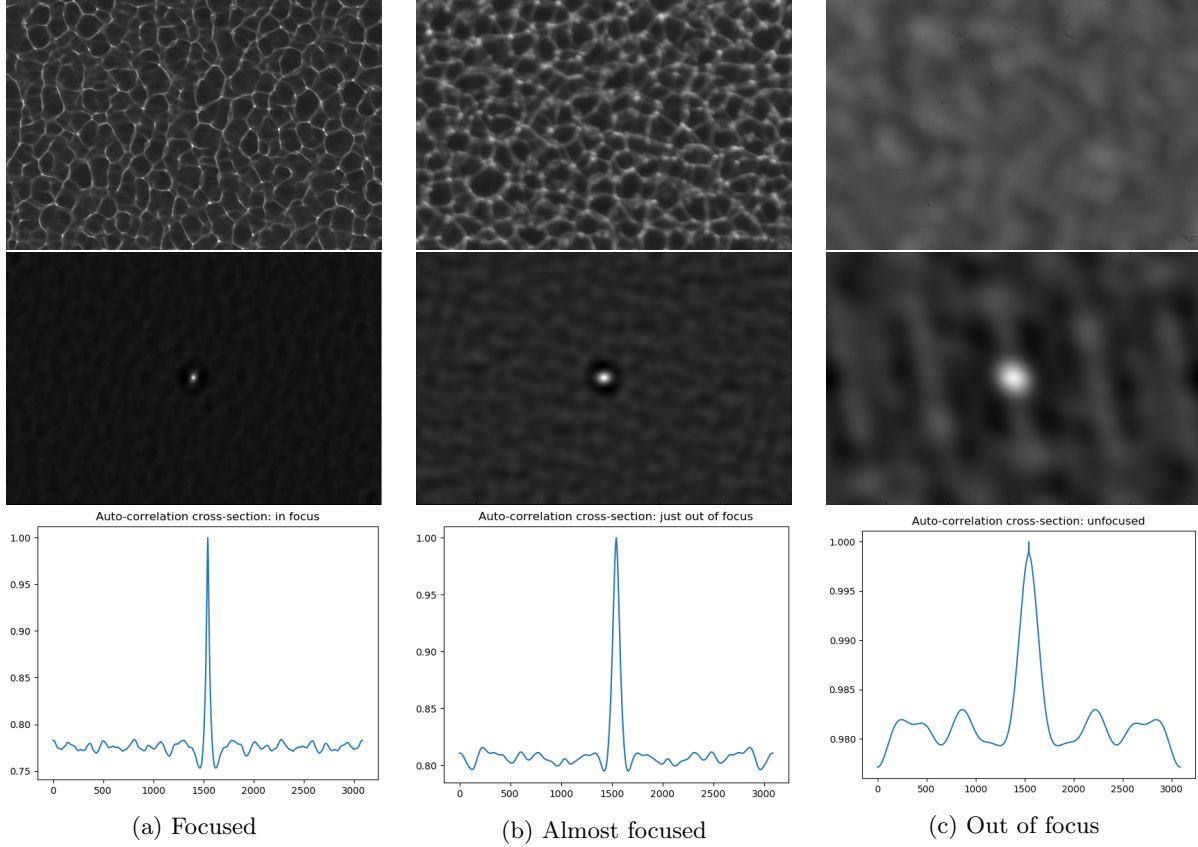


Figure 4: Sensor reading (top) and autocorrelation (middle) of a point light source when (a) the diffuser is placed at the right distance, (b) the diffuser is too far from the sensor, (c) the diffuser is very far from the sensor. Notice that the width central peak of the autocorrelations increase as the diffuser is moved out of focus. This is most evident in the cross-sections (bottom) of the autocorrelations in figures (a), (b), (c). In addition, the surrounding side-lobes in the autocorrelation become more pronounced as the diffuser becomes more out of focus.

## 4 Taking Images

### 4.1 Calibration

Calibration entails a single PSF measurement taken at the same distance as the object you are trying to image. The PSF is depth-dependent: the pattern scales with distance. Moving the light source away from the sensor will result in a smaller pattern. This means that, to properly calibrate the system, we need to record a PSF that is at the same distance as the object of interest. This effect only matters for objects close to the sensor, since the size of the PSF changes noticeably in that range. After a certain distance away from the camera, the change in the PSF is so small that it can be ignored. In this range, the 2D DiffuserCam truly operates on single-shot calibration.

The point source used for calibration should have the right brightness and ideally be the smallest size possible.

1. The brightness should be such that it uses the full dynamic range of the sensor but does not saturate it. In particular, no pixels should have a value equal to their maximum value because then the true brightness at that point is lost. For example, in a typical 8-bit PNG, no pixel should have a value of 255. However, the brightest pixels in the image should still be close to the maximum value allowed by the sensor, because using the full range of pixel values results in a more precise measurement.



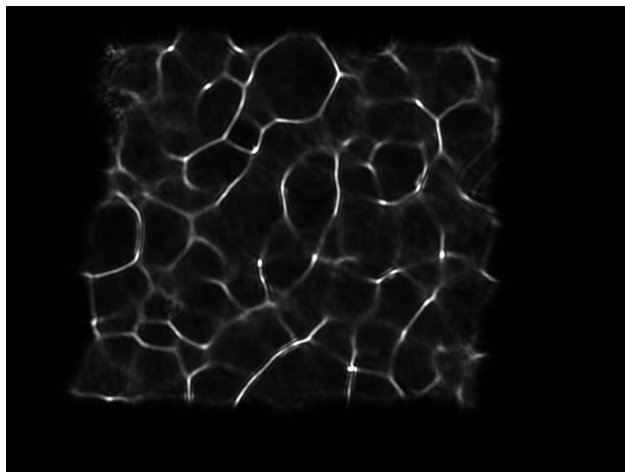


Figure 5: Example cropped psf – a black border around the pattern should be visible. Notice that in Fig. 4 (a) there is no such border.

To address this issue, we attached our LED to a tunable voltage source to make the brightness adjustable. In addition, you can vary the exposure time of the sensor. If your live-view software has an option to display a histogram of the values at each pixel, that is an excellent way to verify that all the pixels in your image are unsaturated but use the dynamic range available.

2. If the light source is not point-like, then the PSF measurement will have blurred lines, which decreases the attainable resolution of the reconstruction.

For our LED, we placed a pinhole aperture made of cardboard in front of it.

## 4.2 Taking actual images

*Figure out the appropriate object distance.* (See Section 5.1) Estimate the minimum feature size that you can resolve by a quick check: move the point source laterally at different depths to see how much the PSF shifts. Distinguishing those shifts is what you really care about, so place your object in a region where its features can be distinguished!

*Illuminate your object well!* Make sure the object is sufficiently illuminated (bright lights focused on the object). Stray reflections from other objects can show up as noise or blobs in the reconstructions, so you want to have a dark background if possible. The room itself should be dark, so that the only light that comes through the diffuser is light reflected off the object.

You should first try taking a picture of an image on a cell phone, because the object is self-illuminating. If you're close to the camera, you may have to lower the phone screen brightness to not saturate the sensor. It is possible that your phone has infrared proximity sensors that will interfere with the sensor readings (see an example of a faulty reading in Fig 6). In this case, placing an IR filter on the sensor may help; we recommend finding a display that does not have this problem. Use a black and white image of something high contrast and with large features or shapes. If this works, move to a physical object with a dark/black background so you don't have to worry about stray light as much. With a better sensor, some of these restrictions can be relaxed.

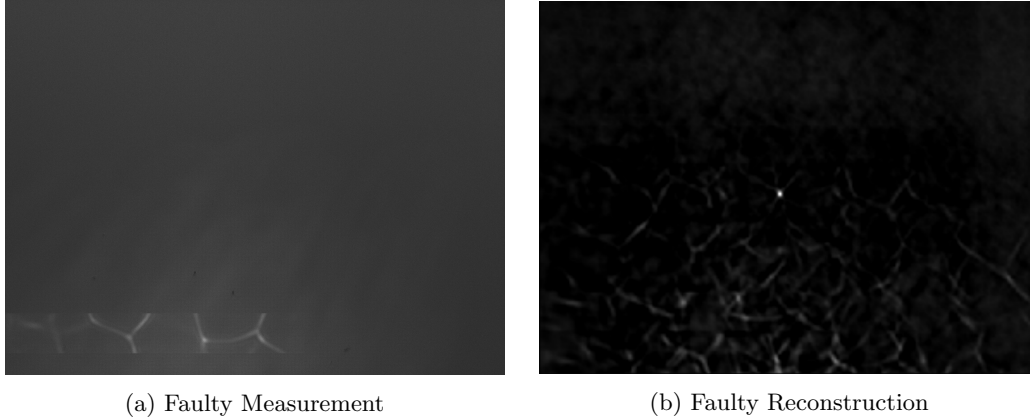


Figure 6: In Fig 6a, one can see a PSF-like block even though the input has no point sources. Typically, the IR sensor produces an artifact like this (where the artifact may move as well). In Fig 6b we can see that the reconstructed result is not correct because the artifact overpowers the rest of the image.

### 4.3 Reconstructing images

These instructions assume an Anaconda distribution because it allows for easy installation of dependencies. If you have your own Python3 development setup, feel free to use that instead. The packages you need to install are listed in the `environment.yml` file.

First, navigate to the directory in which you want to place all the diffuser cam code. Then, enter the following commands:

```
1 git clone https://github.com/Waller-Lab/DiffuserCam-Tutorial.git
2 cd DiffuserCam-Tutorial
3 conda env create
4 source activate diffuser_cam
```

If using windows, use `activate diffuser_cam` instead.

Next, open up the `admm.config.yml` or `gd.config.yml` files and modify the parameters as necessary. Please see the corresponding jupyter notebooks for explanations of those parameters.

Finally, run `python admm.py` or `python gd.py` to reconstruct the images you selected in the config file.

## 5 Analyzing your diffuser

In our experience, double-sided tape was the best diffusive material we found, but you may find that another material better fits your imaging needs (e.g. small objects very close to the sensor might require finer caustics). In this section, we discuss some other options and how to qualitatively analyze those options.

Two things to consider are the focal distance of the diffuser and what the caustic pattern looks like.

### 5.1 Focal distance

The focal distance of the diffuser determines an important tradeoff:

- Larger focal distance corresponds to larger magnification, so small features of your object are more distinguishable.

- Smaller focal distance corresponds to larger field of view (FoV), so you can image larger objects.

The focal distance is proportional to magnification. For an object and sensor placed at distance  $d_o$  and  $d_i$  from the diffuser respectively, the magnification is  $m = -d_i/d_o$ . In our case,  $d_i$  is always the focal distance.

Magnification determines the FoV because of the finite sensor size – at a given object distance, the magnification determines the largest object that can be fully imaged onto the sensor (see Fig 7). The tradeoff above can be explained by this property: decreasing the magnification maps a larger FoV onto the same number of sensor pixels.

You can change the magnification by changing the object distance. So, for an object of fixed size and a setup with fixed focal distance, there’s a “nice” region where the object occupies a reasonable portion of the field of view (i.e. it fits, but it’s not too small, see Fig 7). This is especially important to keep in mind for DiffuserCam because if objects are too far away relative to the focal distance, the reconstruction will look like a point source. In a regular imaging system, we could immediately attribute this effect to low magnification. In DiffuserCam, we can’t check the magnification visually (we can’t interpret the raw data), so the only way to verify that the magnification is appropriate is to calculate it based on the focal and object distances.

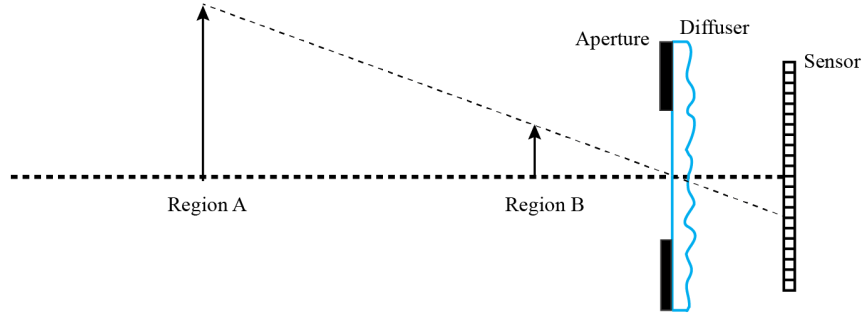


Figure 7: In Region A, larger objects can be imaged than in Region B. With a normal imaging system, it’s easy to find the appropriate object distance, but because reconstructions aren’t instantaneous it’s important to do a rough calculation beforehand.

## 5.2 Caustics

Qualitatively, the caustic pattern itself should have:

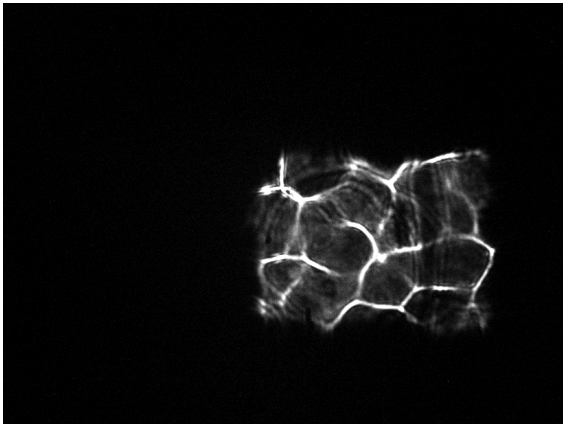
- *Thin lines*, so smaller shifts are distinguishable. One way to evaluate it objectively is to measure the width of the autocorrelation peak.
- *Lines in many directions*, so shifts in arbitrary directions are equally distinguishable. By eye, if the pattern is dense but has randomly oriented lines (i.e. not a grid), it is a good candidate. One way to evaluate it objectively is to see if the width of the autocorrelation peak is the same in every direction.

Once you place the aperture on the diffuser, the imaging system should be *linear shift-invariant*. Shift-invariance means that a lateral shift of the point source causes a lateral translation of the caustic measurement. Linearity means that scaling the intensity of a point source correspond to scaling the intensity of the sensor reading by the same amount. Also, the pattern due to two point sources is the sum of their individual contribution. In addition, linearity may depend on the sensor – some sensors may not respond linearly to increasing light intensity. You should test the shift-invariance and linearity of your system by shifting and adjusting the brightness of your point source.

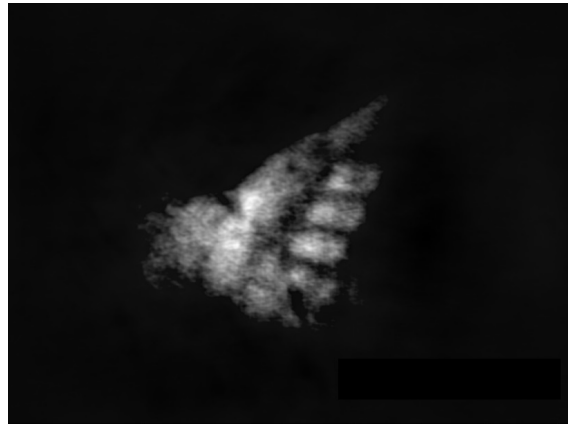
### 5.3 Examples

Here are a few diffusers that we tested, to give an idea of the tradeoffs involved. The “degree” of a diffuser refers to the average angle that a light ray bends when it goes through the material. So a higher degree diffuser generally has a smaller focal length. If you want to test machined diffusers that have specific scattering properties, we found ours here: <https://www.luminitco.com/>.

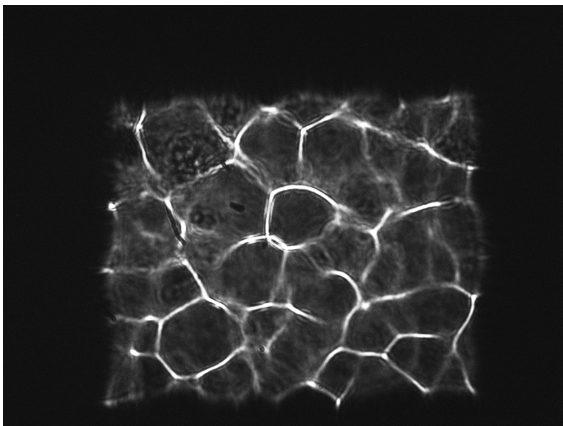
- 0.5 degree – large focal distance ( $\sim 5$  mm), so the field of view is small and magnification is large. The caustic density is low, so if the aperture is too small there may not be enough to get a good reconstruction.



(a) PSF, with a large crop due to the small aperture



(b) Reconstructed image of a hand, 40cm away

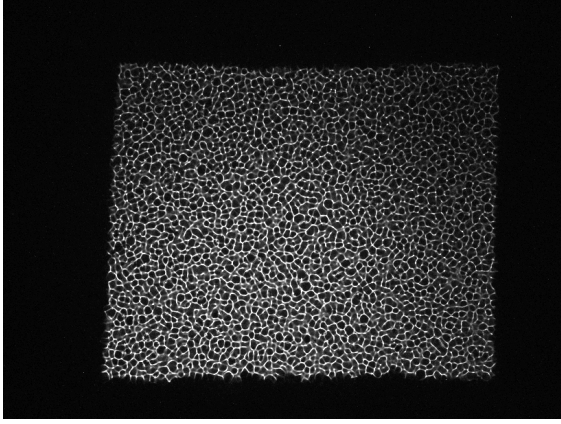


(c) PSF with appropriately sized aperture

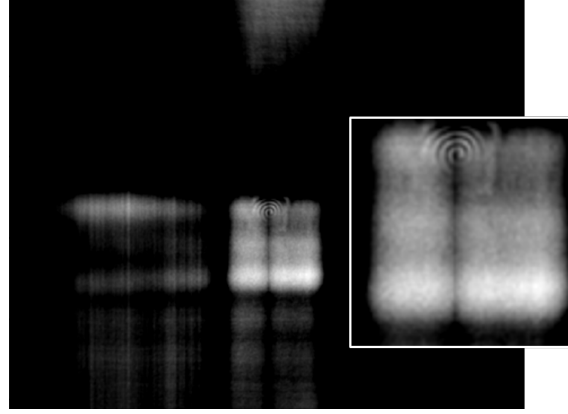


(d) Reconstructed image of a hand, 40cm away. More lines in the PSF results in a better reconstruction

- 1 degree – very small focal distance (the diffuser is basically on the sensor – generally bad for shift invariance), huge field of view, very low magnification. Need to place object super close.

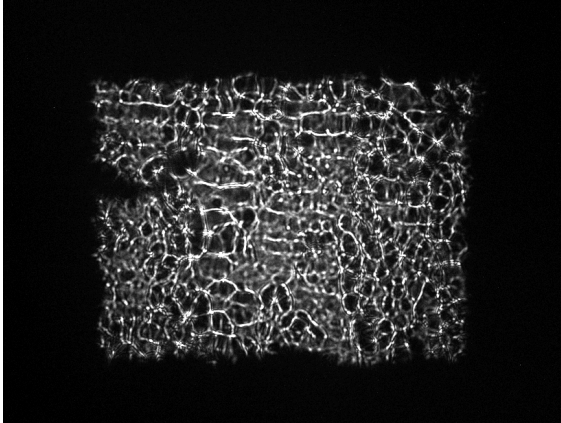


(a) PSF

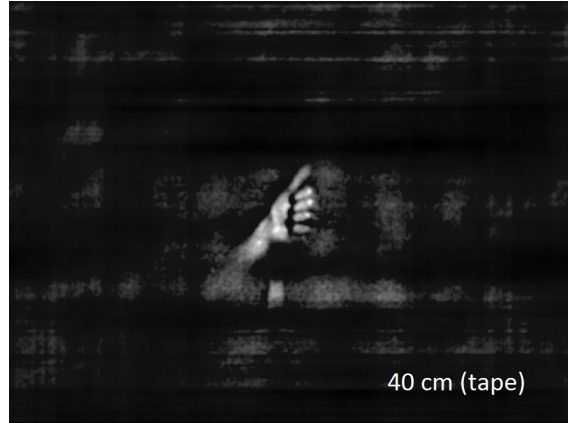


(b) Reconstructed phone image of a spiral, 7cm away (hand not possible)

- Tape – slightly larger focal distance ( $\sim 3$  mm) than 1-degree, so field of view/magnification is more manageable, but more concentrated caustics so reconstruction quality is better than the 0.5-degree.



(a) PSF



(b) Reconstructed image of a hand, 40cm away

Figure 10

## 6 PiCamera Specific Instructions

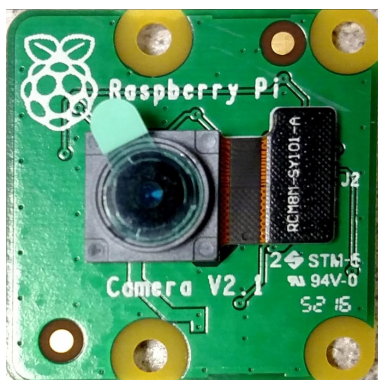
### 6.1 Hardware

The Raspberry PiCamera is a lens screwed in a black plastic mount (Fig. 11a). We only care about the sensor, which can be accessed by unscrewing the lens (Fig. 11b). If the diffuser focal length is smaller than the height of the mount ( $\sim 4$ mm) then you need to pry off the mount itself. WARNING: This is not the intended use! It's very easy to damage the camera, so be careful.

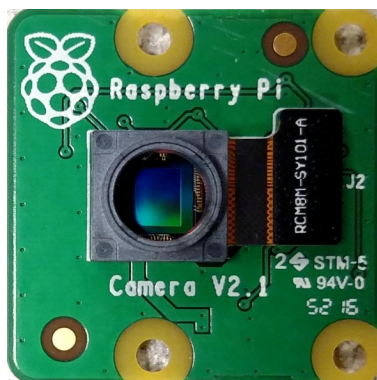
Cut the glue between the mount and the sensor with a blade (Fig. 11d). When cutting, be careful not to damage any of the electrical components on the edge of the sensor – see Fig. 11e to determine which side to start cutting from. One of the sides has fewer components.



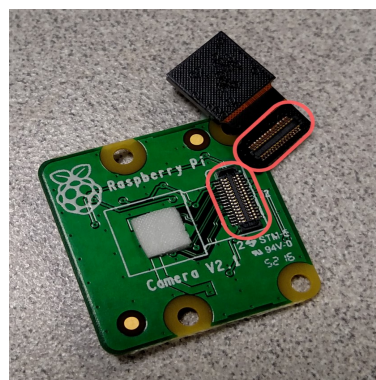
If you want, you can remove the entire sensor-mount component from the board by carefully lifting it off and breaking the glue/pad that connects it to the board. Push the sensor back onto the board once the mount is removed. Make sure the connections from the sensor circuitry to the board are fully clicked in place (Fig. 11c).



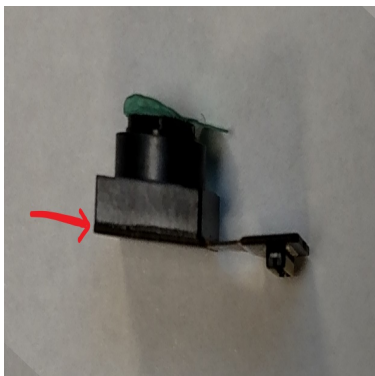
(a) PiCamera with the lens.



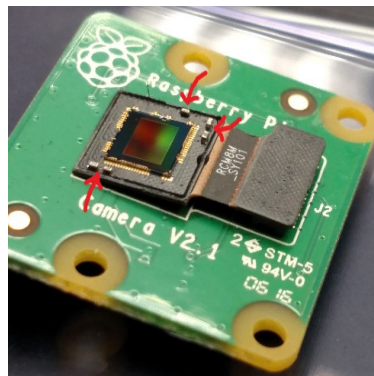
(b) Camera with the lens unscrewed.



(c) Sensor taken off the board.



(d) Sideview of the camera.



(e) Camera with the mount removed. The arrows indicate electrical components to avoid when cutting the glue

Figure 11: The Raspberry Pi Camera

## 6.2 Software

For instructions on how to download the DiffuserCam code and run the algorithms, see 4.3. These instructions are for setting up your raspberry pi and picamera in a convenient way for taking pictures and transferring the images to your computer for processing.

1. [Set up](#) the Raspberry Pi operating system (NOOBS is easiest)
2. If you don't mind always having the raspberry pi and camera connected to an external monitor and keyboard, skip to step 3. Otherwise:
  - (a) Connect the Pi to wifi:
    - [A general guide](#)
    - NOTE: If you need to connect to a network that requires both a login and password (like AirBears2 on Berkeley's campus), follow the corresponding section in [these](#) instructions.



- (b) Set up VNC on your laptop and the Raspberry Pi:
- [Enable](#) VNC Server on your Pi (or download it if your OS isn't standard).
  - [Set up](#) the VNC viewer on your laptop and connect.
  - When you connect to the pi via VNC Viewer, if the display doesn't fill the entire screen you should go to **Preferences>Raspberry Pi Configuration>Set Resolution** and set it to a resolution that allows it to fill the VNC Viewer window. Otherwise, when you use our preview code, the terminal might get obscured by the preview.
- (c) On the Raspberry Pi VNC Server, go to **Options > Troubleshooting** > Check the **Enable experimental direct capture mode** box. This lets the picamera preview display correctly on the VNC Viewer.
3. Install `numpy`, `pillow`, and `picamera`. The raspberry pi is very slow to assemble numpy if you use `pip`, so we recommend (note we're using `pip3` because all our code is Python3):
- ```
1 sudo apt-get install python3-picamera #Installs python3-numpy along with it
2 sudo apt-get install python3-pillow
```
4. Copy the `preview.py` file to your Pi.
- If you set up VNC, then you can use the **Transfer Files** option in the VNC menu at the top of your window.
5. Connect the picamera to the appropriate slot in the Pi (it should be labeled "CAMERA") and restart your terminal.
6. Run the file via `python3 preview.py`. Follow the prompts in the terminal to change shutter speed, take the image, and name the image file.
- With VNC, you can transfer the image back to your computer using the same file transfer option.