Analyzing Light Fall-Off Stereo

Helena Yang
Carnegie Mellon University
hfy@andrew.cmu.edu

Abstract

Depth recovery techniques in computer vision (ex. focus and defocus, shading, textures) often assume that objects in a scene are diffuse or Lambertian, or require precise calibration of cameras and light sources in order to retrieve accurate data. In contrast, depth recovery based on light-fall off intensity (i.e. leveraging the inverse square law for light intensity) allows depth recovery of non-Lambertian objects, and works with an uncalibrated camera. Two methods in Liao et al.*'s Light Fall-Off Stereo paper [1] are used to improve the robustness and simplicity of the depth map recovered for an object (local and global). I implemented both methods and tested them on a series of photos taken with a Nikon D3500 camera and a Kodak Luma 150 projector. Additionally, I tested the effectiveness of both methods under various conditions, including using objects of different materials, different incident light directions for light fall-off depth recovery, and different ranges of light distances from a depth reference plan to be used for processing.*

1. Introduction

Depth recovery techniques in computer vision (ex. focus and defocus, shading, textures) often assume that objects in a scene are diffuse or Lambertian, or require precise calibration of cameras and light sources in order to retrieve accurate data.

For example, photometric stereo is a stereo technique that infers per-pixel normal and albedo information from a series of images where the camera is fixed, but the light source location varies in different locations around the object. The normals are then integrated to produce depth maps. However, Photometric stereo is only designed for lambertian objects with assumed constant albedo, and does not support depth recovery of objects of other materials as robustly. Other depth recovery techniques such as focus and defocus, or techniques that leverage shading and textures, also suffer from similar limitations.

In contrast, light fall-off intensity is a depth cue that applies to non-lambertian materials in addition to lambertian materials, and offers a different stereo perspective than past techniques. Moreover, light fall-off intensity-based stereo does not require a calibrated camera, and can provide reasonable depth map results simply by varying the distance a light source is from an object. For my project, I implemented two light fall-off intensity-based stereo methods from Liao *et al.*'s Light Fall-Off Stereo paper [1], and tested their effectiveness under various scenes, lighting configurations, image processing conditions, and object materials.

2. Local Method

2.1. Light Fall-off Distance

Light fall-off stereo exploits the light fall-off distance property that objects have when illuminated by some light source. Specifically, the *inverse square law* for light intensity states that the brightness of a point on the object's surface is inversely proportional to the square of how far the point is from the light source, i.e.:

$$I(p) = \frac{k_p}{r_p^2}$$

where $I(p)$ is the intensity of the pixel value of scene point p, r_p is the distance the scene point is from the light source, and k_p is a constant that is determined by the object's reflectance and orientation, as well as the light source's intensity.

Since k_p is a constant (measuring intrinsic properties) and does not change when the the light source moves away from the object, it follows that for a light source distance r'_p away from scene point p (where the light source is still oriented in the same direction as the previous scene configuration):

$$I'(p) = \frac{k_p}{r_p'^2}$$

Therefore, if we have two different lighting configurations I, I', then we can relate the squares of the depths from the light sources using the intensities themselves:

$$\frac{I'(p)}{I(p)} = \frac{r_p^2}{r_p'^2}$$

If we also know $\Delta r = r'_p - r_p$, i.e. the distance that the light source moved between the two lighting configurations, then we can compute the depth of p in the first lighting configuration I with our three known values $I, I', \Delta r$:

$$r_p = \frac{\Delta r}{\sqrt{I(p)/I'(p)} - 1} \tag{1}$$

where r_p measures the depth of pixel p from the perspective of the light source at configuration I.

2.2. Scene Depth Approximation

While we could vary the light source's distance for each pixel to compute its depth from a light source using equation 1, in practice this is tedious and prone to errors relating to handling the light source precisely. Liao *et al*. instead uses equation 1 to approximate the depth for the entire scene using only two images [1]. This approximation assumes that the incident lighting directions for different points in the scene are parallel, which is a reasonable assumption if the light source is sufficiently far from the object, and is the basis for the *local* method described in the Light Fall-off Stereo paper– take two images with varying light distances, and compute a pixel-wise depth map using their intensities I, I', as well as the distance the light source moved to get between them Δr.

In my implementation, I took two RGB images of a scene taken at different lighting distances, linearized them and converted them to XYZ space, extracted their luminance channels, and computed equation 1 for all image pixels using vectorized `numpy` operations. To prevent division by zero errors and `nans`, I added a small epsilon to the denominator in equation 1. I also normalized the resulting depths for visualization.

3. Global Method

Although the local method is efficient and simple to implement, Liao *et al*. describes how the depth maps generated by it are often noisy due to camera or scene noise. As a result, the paper also describes a different *global* light fall-off stereo method that optimizes for smooth and accurate depth maps by incorporating image from more than two lighting configurations and enforcing spatial smoothness in the optimized depth map.

3.1. Multi-image Energy Minimization

Liao *et al*. observes how the depth of a pixel from the point of view an arbitrary light source of distance Δr_i away from the closest light source's measured depth $r_{x,y}$ can be expressed in terms of the $r_{x,y}$ and the intensities of the arbitrary light source's pixel $I^i_{x,y}$ and the closest light source's pixel $I_{x,y}$ (i.e. the depth reference plane's corresponding image's measured intensity of the pixel at (x, y)):

$$\sqrt{I_{x,y}} r_{x,y} = \sqrt{I^i_{x,y}} r^i_{x,y} = \sqrt{I^i_{x,y}}(r_{x,y} + \Delta r_i)$$

In particular, if we have N different configuration, then in an ideal setting we expect to be able to relate their intensities to one another through $r_{x,y}$ and Δr_i:

$$\sqrt{I_{x,y}} r_{x,y} = \sqrt{I^1_{x,y}}(r_{x,y}+\Delta r_1) = \cdots = \sqrt{I^N_{x,y}}(r_{x,y}+\Delta r_N) \tag{2}$$

In practice we may not be able to find $r_{x,y}$ that satisfies equation 2 due to camera and image processing noise. The next best thing to do is to find $r_{x,y}$ depth values that minimize the variance among the terms in equation 2, which is the following energy objective E'[1]:

$$E' = \sum_{x,y} \sum_{i=0}^{N} (K_i - \bar{K})^2 \tag{3}$$
$$K_i = \sqrt{I^i_{x,y}}(r_{x,y} + \Delta r_i), 0 \le i \le N$$

where K_i represents the ith term from equation 2, and $\bar{K}$ is the average of all K_i terms.

Liao *et al*. also incorporates a second penalty term E'' into the objective function to enforce smoothness between adjacent pixels in the depth map:

$$E'' = \sum_{x,y} (u^2_{x,y} + v^2_{x,y}) \tag{4}$$

where u, v are symmetric finite differences of $r_{x,y}$ [1].

$$u_{x,y} = r_{x+1,y} - 2r_{x,y} + r_{x-1,y}$$
$$v_{x,y} = r_{x,y+1} - 2r_{x,y} + r_{x,y-1}$$

The final energy minimization objective function is a linear combination of equation 3's variance penalty E' and equation 4's smoothness term [1]:

$$\begin{aligned} E &= (1-\lambda)E' + \lambda E'' \\ &= (1-\lambda) \sum_{x,y} \sum_{i=0}^{N} (K_i - \bar{K})^2 + \lambda \sum_{x,y} (u^2_{x,y} + v^2_{x,y}) \end{aligned} \tag{5}$$

where $0 \le \lambda \le 1$ is a weight determined by the user.

3.2. Optimization Approach

Liao *et al*. solved for the optimal $r_{x,y}$ values that minimize E in equation 5 using Conjugate Gradients and line search, where they imposed boundary conditions requiring the gradient of E with respect to the vector of all pixel depth values $\vec{r}$ to be 0 at boundary pixels [1]. The paper did not provide any implementation details beyond this description.

In my implementation of the global method, I first expressed equation 5 as a quadratic form:

$$E = \frac{1}{2}x^T Ax - b^T x + c \tag{6}$$

where x, b are vectors and A is a matrix. If the depth image we are solving for has dimensions $n \times m$, then A is a $(n+2)\times(m+2)$ by $(n+2)\times(m+2)$ matrix, and x, b each are size $(n+2)\times(m+2)$. The extra $+2$ in each dimension accounts for padding on the boundary of the actual depth image pixels, in order to impose the same boundary conditions as in the original LFS paper. (I treat x as a padded solution to $\vec{r}$ to solve for.)

In this quadratic form, the x that minimizes E is the x that is a solution to $Ax = b$ [2]. Therefore, after forming A and b, as well as a mask for the non-boundary pixels in x, I run standard conjugate gradient descent [2] on A and b to solve for x, whose solution I can reshape to visualize as a depth map.

I represent x as a flattened padded depth image to solve for, so all cells in x are ordered in row-major order.

To form A and b, I first added in the relevant terms for the variance penalty E'. In particular, for a single pixel's depth $r_{x,y}$, the penalty for a single image i $K_i - \bar{K})^2$ can be expressed as:

$$(ar_{x,y} + c)^2$$

$$\text{where } a = \sqrt{I^i_{x,y}} - \frac{1}{N}\sum_{j=0}^{N}\sqrt{I^j_{x,y}}$$

$$\text{and } c = \sqrt{I^i_{x,y}}\Delta r_i - \frac{1}{N}\sum_{j=0}^{N}\sqrt{I^j_{x,y}}\Delta r_j$$

Expanding out $(ar_{x,y} + c)^2$, we have:

$$a^2 r^2_{x,y} + 2acr_{x,y} + c^2$$
$$= \frac{1}{2}(2\cdot a^2)r^2_{x,y} - (-2ac)r_{x,y} + c^2$$

Therefore I added $(1-\lambda)\cdot 2\cdot a^2$ into $r_{x,y}$'s diagonal cell in A, and $(1-\lambda)\cdot -2ac$ into $r_{x,y}$'s cell in b to account for the variance term caused by image i, and did this for each image I considered, for every pixel.

To account for the smoothing term, I added in $u^2_{x,y}$ and $v^2_{x,y}$ terms in for each pixel separately into A. To determine what values to place into the cells of A, I expanded out each smoothing term. For example, $u^2_{x,y}$ expands out to:

$$(r_{x+1,y} - 2r_{x,y} + r_{x-1,y})(r_{x+1,y} - 2r_{x,y} + r_{x-1,y})$$
$$= r^2_{x+1,y} - 2r_{x+1,y}r_{x,y} + r_{x+1,y}r_{x-1,y}$$
$$+ -2r_{x,y}r_{x+1,y} + 4r^2_{x,y} - 2r_{x,y}r_{x-1,y}$$
$$+ r_{x-1,y}r_{x+1,y} - 2r_{x-1,y}r_{x,y} + r^2_{x-1,y}$$

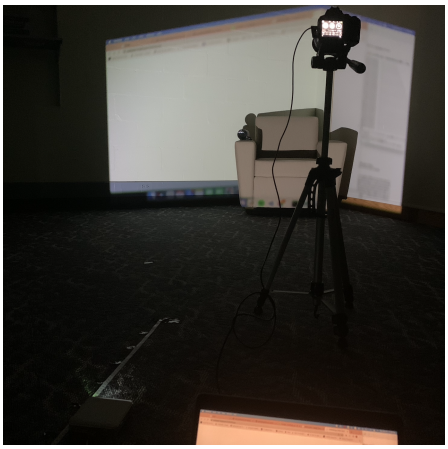


Figure 1. Hardware setup with camera, projector, tape measure.

To get an equivalent expansion from an Ax multiplication, I added in $\lambda\cdot 1$ into the cell in A corresponding to $r_{x+1,y}$'s row and col, $\cdot -2$ into the cell in A corresponding to $r_{x+1,y}$'s row and $r_{x,y}$'s col, $\lambda\cdot 1$ into the cell in A corresponding to $r_{x+1,y}$'s row and $r_{x-1,y}$'s col, etc.

4. Experiment and Results

4.1. Photo and Hardware Setup

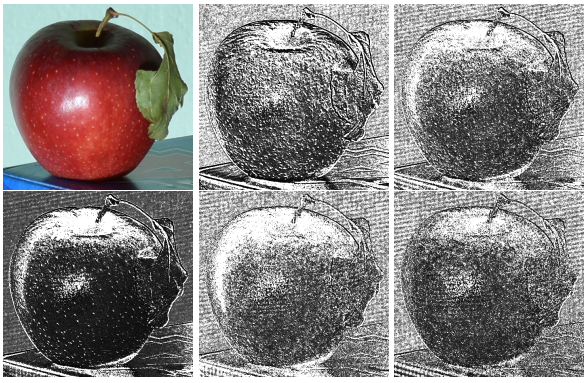
To collect photos for testing my local and global methods, I fixed a Nikon D3500 camera on a tripod in a dark room, and took photos remotely on my laptop using `gphoto`. I used the lowest ISO setting (100) and a constant shutter speed that would avoid over-saturated and under-saturated pixels as best as possible under the lighting conditions in the room. For my light source, I used a Kodak Luma 150 projector, which I also connected to my laptop to shine a white light on my target object. To simulate the moving projector in the original LFS paper, I moved my projector along a tape measure fixed on the ground at certain increments, so I could both preserve the lighting direction at different distances, and also measure the distances between different lighting configurations.

I gathered photos at 16 different distances for objects representing a variety of materials, such as mostly lambertian (banana), translucent (water bottle, soap dispenser), and slightly glossier/specular materials (apple, umbrella, mug). For the apple and banana, I also captured varying light distance photos from different directions around the object (ex. in the same orientation as the camera, slightly to the left of the camera but facing the object, to the right of the camera but facing the object).

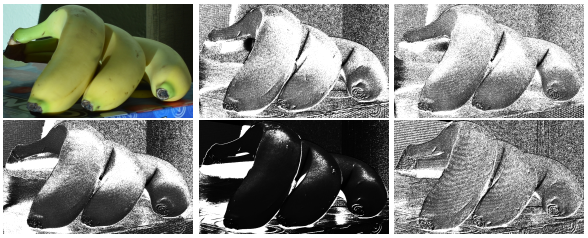
4.2. Local Method Results

For the following visualizations, (i,j) indicates that the depth map was formed using images where the projector

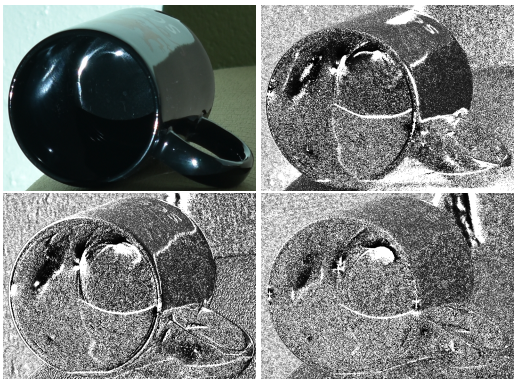
was i and j cm away from the reference depth plane, where the reference depth plane refers to the configuration where the projector is the closest to the object.



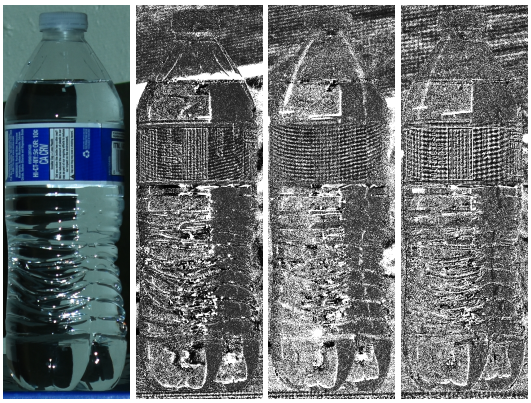
Row 1 left to right: photographed apple, (0,10), (10, 20)
Row 2 left to right: (10,30), (30,40), (40,50)



Row 1 left to right: photographed banana, (0,10), (10, 20)
Row 2 left to right: (40,60), (50,150), (100,110)



Row 1 left to right: photographed mug, (0,10)
Row 2 left to right: (50,60), (100,110)



Left to right: photographed water bottle, (0,10), (40,50), (100,110)

The local method gave reasonable depth map results for mostly or partially lambertian objects such as the apple and the banana, although the quality of its per-pixel depths seem to depend a lot on which distances (i,j) are picked to form the map. In particular, picking distances that are very close to the object (such as (0,10)), often induces stronger artifacts in the depth map. This is likely due to the fact that closer distances are more likely to violate the scene depth approximation from **Section 3.2** more noticeably.

Additionally, while larger values of (i,j) can sometimes give more reasonable depth maps, extremely large (i,j) values tend to generate depth maps with less contrast. This is likely due to the fact that at further distances, it is harder for the camera to pick up significant differences caused by light fall-off distance (as it decreases rapidly by a square factor).

During my experiments, I found that the most reasonable depth maps also used close values of i and j, i.e. if i and j were far apart (such as (50,150) for the bananas), the depth maps were likely to have a lot more noise and artifacts than closer i,j values. This may be due to the fact that lighting directions per pixel change less for smaller smaller changes from i to j, and so smaller (i,j) intervals violate the scene depth approximation less.

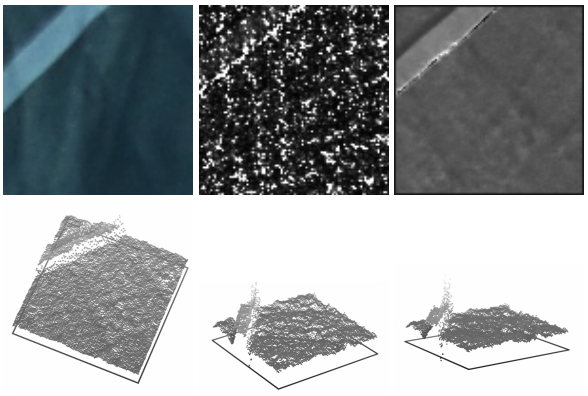
Under certain configurations, the local method was also able to approximate the depths of some parts of specular objects, such as the mug under its (100, 110) configuration. However, it seems that the local method still suffers some similar problems as photometric stereo in that it can't detect the depths of regions containing strong specularities, shadows, or translucent objects very accurately.

Overall, the depth maps generated from the local method all tend to be very noisy, which the global method aims to address by considering more than one image and by enforcing spatial smoothness.

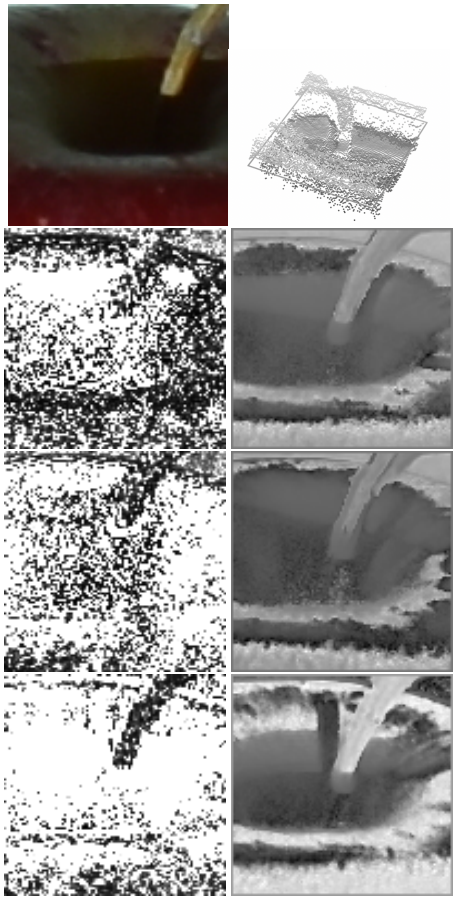
4.3. Global Method Results

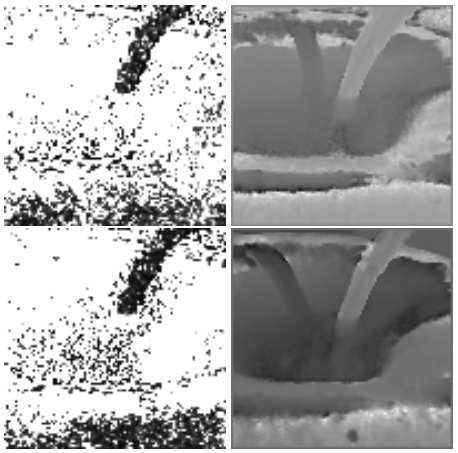
Because the A matrix I formulated in **Section 4.2** is very large for large images (n by n for n pixels in an image),

I was unable to test this method on larger images without running out of memory, Python crashing, or my program being unbearably slow. As a result, I only tested the global method on smaller, 100 by 100 pixel images. I typically used smoothing factor λ values of $0 - 0.15$, and ran conjugate gradient descent with a maximum of 4000 iterations, an ending epsilon of 0.001, and an initial x of all 0's.

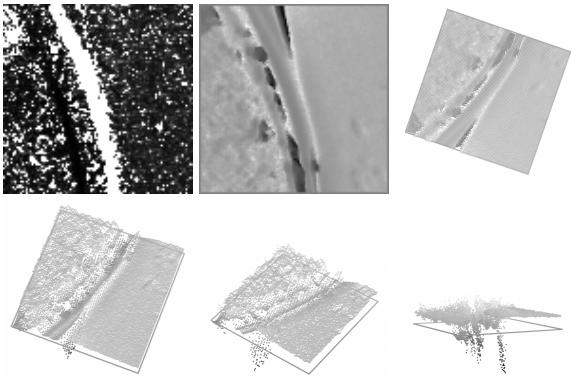


Row 1 left to right: photographed umbrella fabric, local method with (30,40) distances, global method with $\lambda = 0.15$
Row 2: Depth maps for the global method





Row 1 left to right: photographed apple section, 3D depth map
Rows 2-5 left to right: local method (0,10), global method under different lighting orientations, $\lambda = 0.15$



Row 1 left to right: mug edge local method depth map with (0,10) distances, global method with $\lambda = 0.15$, 3D depth map
Row 2: More depth maps of the global method

Overall, the global method generated less noisy and much smoother results than the local method. However, it was also much more costly to compute, often taking between 1000-4000 iterations of conjugate gradient descent to converge for these 100 by 100 pixel images.

I found that any artifacts in the global method also depended on the lighting direction chosen for the series of images to be processed, as that affected the existence of any shadows or strange specularities in the resulting depth map.

Finally, the global method is able to detect depths semi-correctly for more glossy objects such as the mug, but it still suffers from existing problems in photometric stereo and the local method, such as extreme specularities and shadows.

5. Discussion

My local method results seemed to be noisier than the local method results from the LFS paper, which may be due to their hardware setup being different than mine. I may have violated the scene depth approximation in **Section 3.2**

more severely than they did because I wasn't able to tilt my projector upwards at an angle in a consistent way.

It's unclear to me how exactly the original LFS paper implemented their global method efficiently, since they were able to work with larger images than I was. I'm unsure whether their line search would have sped up my own optimization process greatly, since I would have still had to work with large matrices, or if I maybe just formulated my optimization problem inefficiently.

Future extensions that would be interesting to explore would include somehow incorporating depth information from multiple directions of light fall-of-stereo into one robust depth map (i.e. taking more inspiration from photometric stereo). It may also be possible to explicitly incorporate BRDF information into the light fall-off stereo global method pipeline (i.e. into the A or b matrices) to yield more accurate results for specific materials.

References

[1] M. Liao *et al*. Light fall-off stereo, 2007. 1, 2

[2] J.R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain, 1994. 3