# Ray tracing modeling and inversion of light intensity under a water surface

*Abdullah Al Theyab*

## ABSTRACT

Light propagating in a water-filled pool is perturbed by the water surface, creating patterns on the pool floor. In this report, I use ray tracing to compute an approximation of the light intensity field on the pool floor using point source and exploding surface models. The ultimate goal is to infer the water surface from the intensity field. In a seismic imaging, it is similar to imaging using amplitudes rather than travel-times. I present a geometric approach to invert for a discretized surface from a ray-count representation of the intensity field. With this formulation, the inversion becomes a combinatorial problem, which can be solved using non-deterministic search techniques. The formulation has a large inherent null space. The low cost of the technique allows a large number of iterations to be applied.

## INTRODUCTION

Light propagation is similar in both nature and theory to seismic wave propagation in the subsurface. It is not surprising to see a large number of common problems between the fields of seismology and optics. Claerbout (2007) poses a question about the relationship between light patterns under water and seismology. One prominent problem in reflection seismology is estimating seismic velocities in the heterogeneous subsurface. Velocity is a measure that depends on travel time; i.e. it is determined by the moveout of seismic events. It is always costly to estimate subsurface velocities, whether using conventional velocity analysis or inversion methodologies. Unfortunately, imaging algorithms rely on the accuracy of velocity models. Therefore, a curious scientist might ask: *Can we use amplitudes alone for imaging?*

For the sake of simplicity, let us consider a simple imaging problem involving a single wavefield caustic: a single reflector including a syncline. Using the exploding reflector concept, we can simulate a zero-offset section (Claerbout, 1985). What we will observe in the section is a bowtie that is caused by the syncline. The syncline causes many ray paths between the exploding reflector to the receiver to converge at some point before reaching the surface, forming a caustic. Given the correct velocity model, migration algorithms like Kirchhoff migration will resolve the bowtie into a syncline (Yilmaz, 2001).

Now, let us consider a swimming-pool experiment where the goal is to infer the water surface from the light patterns on the floor of the pool like the ones shown in Figure 1. We might encounter similar difficulties as in seismic imaging. It is hard to estimate the exact speed of light in the pool. In this case the question is as follows: *Can we use the light intensity field at the bottom of the pool to infer the surface of water?*

The two questions are closely related, and the pool experiment is easier to comprehend intuitively. In this paper, I implement the forward simulation using ray theory to build a light intensity field at the pool floor that can be used for inferring the surface. I will not follow the exact physics as it is done in optical simulations: I assume infinite frequencies for ray theory and representing light with a finite number of rays. The forward simulation is done using ray tracing and beam tracing. I finally discuss a Monte Carlo ray tracing inversion that is based on ray counting.

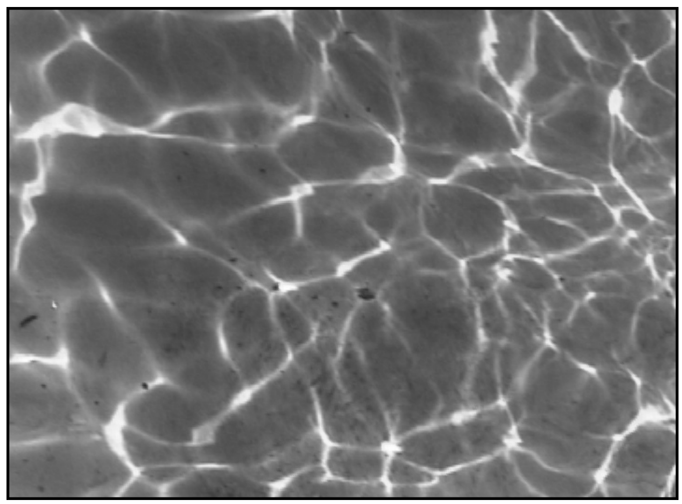

Figure 1: Light patterns at the bottom of a pool (Claerbout, 2007). [**NR**]

## FORWARD SIMULATION

I simulate the light intensity field at the pool floor using ray tracing. For our purposes, we assume that light travels with an infinite velocity; i.e. any ray refracting through the water surface will project instantly onto another point on the floor. If multiple rays for any reason converge at a point before reaching the floor, they form a refracted caustic (Shah et al., 2007).

This simplistic view of the experiment has received considerable attention in the field of computer graphics. Shah et al. (2007), for example, introduce a real-time technique for rendering caustics from reflective and refractive surfaces. Ray tracing techniques have a major deficiency, in that we need an infinite number of rays to simulate a realistic distribution of light. This shortcoming is discussed by Watt (1990) who suggests using beam tracing as an alternative. The vast majority of the techniques developed in computer graphics, however, are developed for a 3D space having point light source(s), polygonal objects, and a single observation point.

Our simplified case inherits only a minor subset of the wide range of techniques used in computer graphics. To start, we will not have an observer point, and therefore do not need to ray trace from the pool floor to the observer point. Also, we have only a single smooth surface of moderate relief, and the incident light direction is limited to rays arriving from above the surface. In other words, we have an unique refraction from each point on the surface. Because we have moderate topology in the water surface, we will assume that the refracted rays do not re-intersect the water surface.

## Surface Representation

In $(x, y, z)$-space, the surface of the water is given as a discretized function on a uniformly sampled grid in one or two dimensions. The function has points that represent the depth of the surface from the $xy$-plane. If we have a surface $S = f(x, y)$ in 3D space, the normal vector to that surface is

$$\vec{n} = \frac{-\frac{\partial S}{\partial x}\hat{\mathbf{i}} - \frac{\partial S}{\partial y}\hat{\mathbf{j}} + \hat{\mathbf{k}}}{\sqrt{(\frac{\partial S}{\partial x})^2 + (\frac{\partial S}{\partial y})^2 + 1}} \quad , \tag{1}$$

and for a surface $S = f(x)$, in 2D space, it is

$$\vec{n} = \frac{-\frac{\partial S}{\partial x}\hat{\mathbf{i}} + \hat{\mathbf{k}}}{\sqrt{(\frac{\partial S}{\partial x})^2 + 1}} \quad . \tag{2}$$

where $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$, and $\hat{\mathbf{k}}$ are unit vectors along the $x$-, $y$-, $z$-axes, respectively. To compute the first partial derivatives, we can use finite-divided-difference approximations (Chapra and Canale, 2002). A design decision must be made at this point regarding where the refracted rays will start with respect to the points on the surface function. For a 2D simulation with $S(x)$, we can have the ray coming out of the points. In that case, the preferred scheme is centered finite-divided-difference:

$$\frac{\partial}{\partial x} S(x_i) \approx \frac{S(x_{i+1}) - S(x_{i-1})}{2\Delta x} \quad . \tag{3}$$

The same differencing scheme can be used for the partial derivatives with respect to $x$ and $y$ for the 3D simulation. Using this scheme, it is not possible to compute the partial derivatives for the edges of the surface, since more points are needed for the computation.

The second design option is to have the rays launching from segments of the surface between the points in the 2D simulation. For the 3D case, we divide every quad – i.e. square between four adjacent points– using two diagonals. For every possible triangle within a quad, we compute the normal and have a ray starting from the center of the triangle. The advantage of this approach is that we have many more rays than the number of surface samples. For computational stability, the surface must be smooth locally with respect to a segment or a quad.

An easy way to construct a synthetic water surface with a sinusoidal wave with a decaying factor as a function of the distance from the source:

$$S(x, y) = c_0 \ \cos\left(c_2|\vec{d}|\right) e^{-c_1|\vec{d}|} \quad , \tag{4}$$

where $\vec{d}$ is the distance from the source, and $c_0, c_1$, and $c_2$ are arbitrary positive constants to customize amplitudes, decay rate, and angular frequency of the ripples, respectively. Waves originating from several source points can be summed to form a more complex surface. This method of construction is for a time-invariant surface without reflection at the surface boundaries. A more realistic water surface can be obtained using finite differencing of the 2D wave equation. For the sake of simplicity, I use the explicit differencing scheme in the $(x, y, t)$-domain. Figure 2 is the result of finite differencing with three sources that are Gaussian wavepackets of different amplitudes.

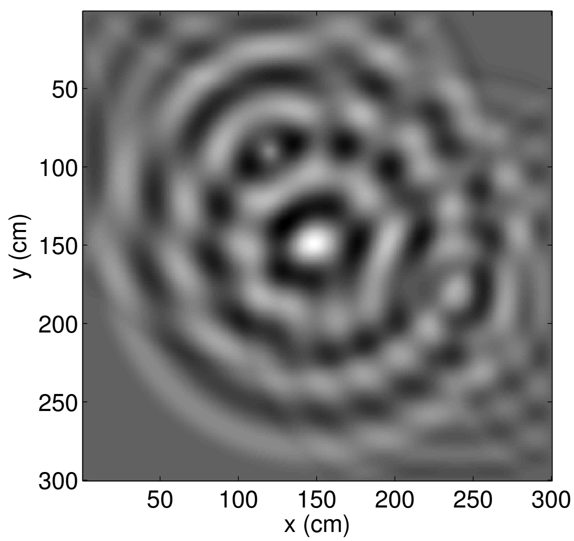


Figure 2: Finite-difference simulation of a water surface with one strong and two weaker sources disturbing the water surface. [**ER**]

## Light Source

The light refracting into the pool can be from a fixed point source from which light is spreading radially, or ambient light scattering from all directions. Usually, the light incident on a real water surface is a combination of both. Because I use rays for modeling light, I classify sources into two categories. The first category includes point sources, from which light is represented as rays traveling to the surface and refracting onto the pool floor (specular-to-specular transport (Watt, 1990)). In the second category, ambient light refracts through the water surface and forms rays (diffuse-to-specular transport (Watt, 1990)). I approach the second case with the notion of an exploding surface.

For a point source, I start by computing the incident ray $\vec{s}$ on each refraction point on the surface. Snell's law can be used in the plane containing both $\vec{s}$ and $\vec{n}$ to find the refracted ray $\vec{r}$ into the water. First, however, the incident ray must be checked to insure that it is not coming from below the surface. This occurs when the source is located very close to the surface creating shadow zones behind high water ripples where there are no rays incident on the surface. Figure 3 shows the refraction of rays into water as well as some shadow zones.

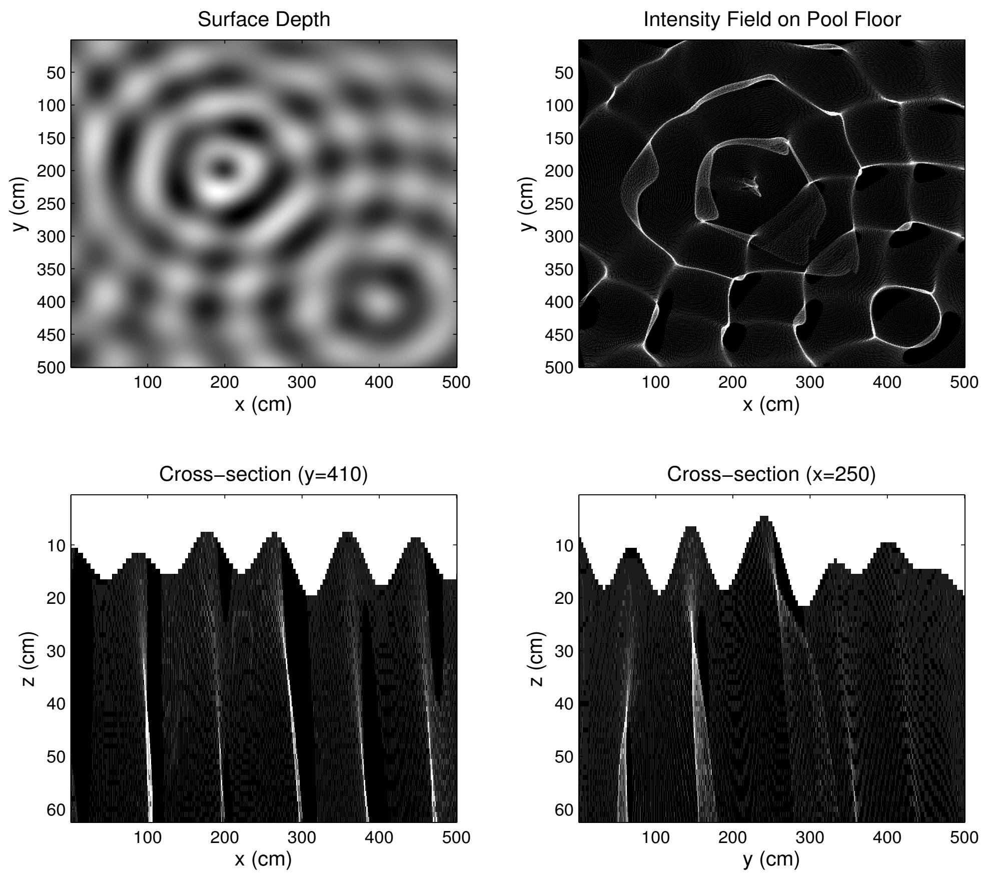


Figure 3: 3D ray tracing simulation with a point source. [**ER**]

As mentioned earlier, I use the exploding surface concept for the diffuse-to-specular mechanism of light transport through the surface. This concept is analogous to the exploding reflector concept that Claerbout (1985) uses as an introductory model for imaging. The refracted ray from each point on the exploding surface travels in the same direction as the surface normal at that point; i.e. $\vec{r} = \vec{n}$. This is an easier approach to the modeling than working with a point source, but it is less accurate.

## Ray Representation

Having developed representations of the surface and of the light entering the surface, the next step is mapping the refracted rays from the surface to the pool floor. At this stage, a ray is simply a normalized vector $\vec{r}$ at a surface position pointing downward. This vector controls the contribution to the light intensity field. The contribution can be a simple binned ray or a beam.

It is a simple exercise of trigonometry to project the refracted ray $\vec{r}$ to a point $\vec{p}$ the bottom of the pool.

$$\vec{p} = \frac{\text{depth}}{r_z} \vec{r} \tag{5}$$

The point $\vec{p}$ is unlikely to fall on a grid point, and therefore the ray amplitude distribution can be binned. If each ray is given a unit amplitude, the sum of amplitudes is the ray count for each bin. Figure 4 shows the result obtained using binning. The observable caustics have sharp, though not continuous boundaries. Zones of low light intensity can be confused with shadow zones, simply because the light is not distributed continuously. Some of the image degradation might be attributed to the nearest-neighbor interpolation; however, even with the best interpolation algorithms, the result will not improve significantly. In Figure 3, the aliasing is due to an insufficient number of rays and binning. de Ridder (2008) suggests using a Gaussian distribution as a function of the lateral distance from the ray projection point on the pool floor. The proposed method yields more stable results. Nevertheless, the symmetry of the distribution neglects the tilting of rays.

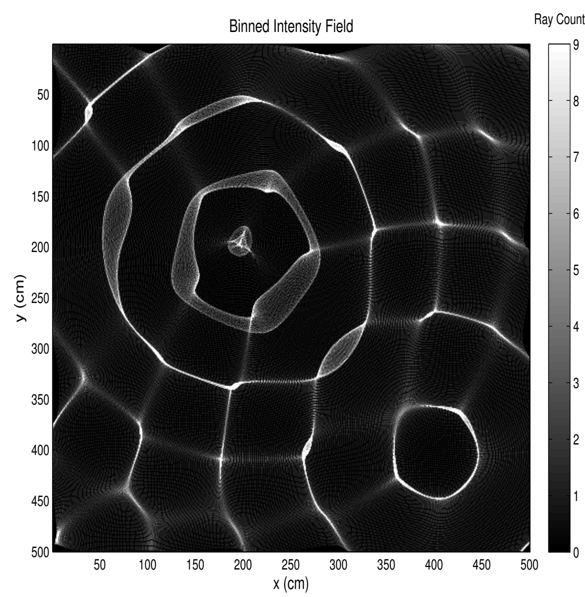


Figure 4: The result of binning is a good approximation but not realistic. [**ER**]

So far I have used the specular-transport mechanism within water and ignored light diffusion, which is a shortcoming of the ray-tracing. It would take orders of magnitude more rays for ray tracing to simulate diffusion, which is prohibitively expensive. Also, it will be inefficient, because diffusion does not change much from one pixel to the next (Watt, 1990). Figure 5 shows the diffusion of light incident from a point on a disordered medium (ICMM, 2008). It is obvious that light intensity within a beam is dependent on the angle of refraction. It can be modeled with a normal distribution as a function of the angle, with a decay factor that is a function of distance from the refraction point. In vector notation, the contribution of the diffusion to a point is

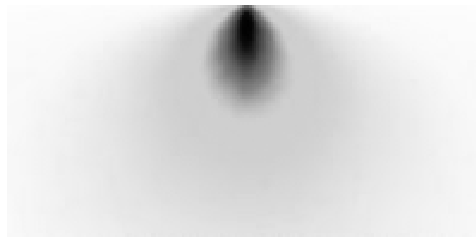

Figure 5: Diffusion of a light beam in an isotropically disordered medium (ICMM, 2008). [**NR**]

$$\rho(\vec{p}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{ -\frac{(|\vec{r}\times\vec{p}|)^2}{2\sigma^2|\vec{p}|^2} - \alpha|\vec{p}| \right\} \quad , \tag{6}$$

where $\vec{p}$ is the position vector from the refraction point $(x_0, y_0, z_0)$ on the surface that is defined as

$$\vec{p}(x,y,z) = (x-x_0)\hat{\mathbf{i}} + (y-y_0)\hat{\mathbf{j}} + (z-z_0)\hat{\mathbf{k}} \quad , \tag{7}$$

$\sigma$ determines how narrow the distribution is, and $\alpha$ is the rate of exponential decay.

Beams[1] demonstrate some of the characteristics of both specular and diffusive transport mechanisms. One way of handling diffusive and specular transport is to model them separately and stack the results. However, I use beams to model the two simultaneously.

In designing the beam shape, one can choose to favor either the diffusive or specular distribution. Favoring diffusion by using equation 6 can negatively affect the resolution of the light patterns. The following equation can be used to describe the distribution of light within the beam:

$$\rho(\vec{p}) \quad = \quad \frac{1}{b} \exp\left\{ -\frac{|\vec{r}\times\vec{p}|}{2b} - \alpha|\vec{p}| \right\} \quad , \tag{8}$$

where $b$ controls the width of the beam, and $\alpha$ is the rate of exponential decay. Figures 6 and 7 show the beam coming from a single point on the surface and the section obtained by stacking all the beams coming from the surface respectively.

The disadvantage of using beam tracing is that every beam contributes to a very large number of points, which can be computationally more expensive –$O(N^2)$– than

[1]Beams discussed here are different from beams used in seismic imaging.

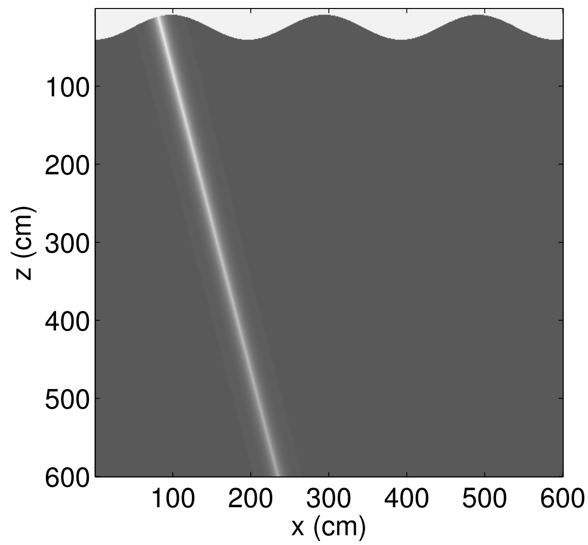


Figure 6: A single beam refracting into the pool. [**ER**]

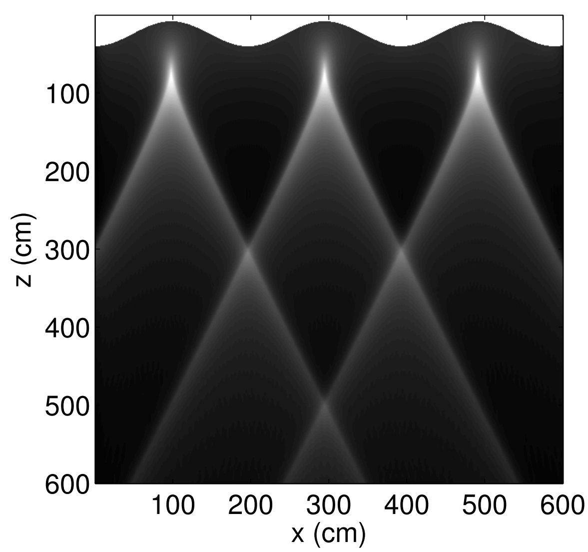


Figure 7: 2D forward modeling using beam tracing. The refracted light focuses into caustics. [**ER**]

binning–$O(N)$, where N is the number of samples on the surface function. As mentioned earlier, the boundaries of the caustics are not as sharp. However, they become more realistic, as in Figure 1, where the details of caustics are not usually seen with sharp boundaries. Figure 8 is the result of the 3D beam tracing of the surface in Figure 2. Figure 9 shows the light intensity field of a surface distorted by many ripples. Although it is more expensive, beam tracing works better than ray tracing. We can distinguish shadow zones and the elongated caustics that usually connect two point caustics.

Beam tracing has many of the limitations of ray tracing with binning. The prominent limitation is that beam tracing needs a large number of beams to model light intensity under a very narrow trough on the surface. Also, for depths where many rays do not project on the pool floor, there are too few beams to construct the correct intensity field. Moreover, the quality of the forward modeling results is sensitive to the beam width chosen; too wide beam hinders the resolution, and too narrow beams behave like rays and can produce aliased results.

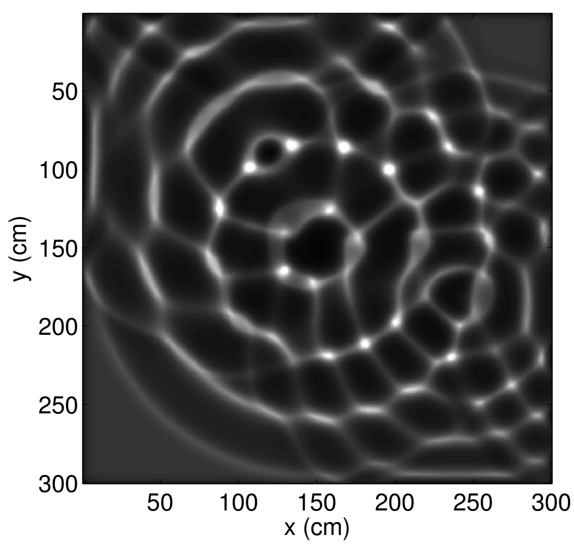


Figure 8: Intensity field under the surface at Figure 2. [**CR**]

## RAY-TRACING INVERSION

As shown earlier, ray tracing with binning can give a quick but crude approximation of the light intensity field at the pool floor. If we have a ray count for light intensity created using the exploding surface model, we can use a simple geometric approach for inversion. In two dimensions, the ray count in a single bin corresponds to the number of surface segments whose normal vectors point to that bin. If no rays hit the sidewalls of the pool, the surface can be reconstructed by assigning each surface segment to one unit of ray count in the intensity field; every assignment means the starting and the ending points of a refracted ray. However, there are many permutations of assignments that can produce the same intensity field. With this formulation, the

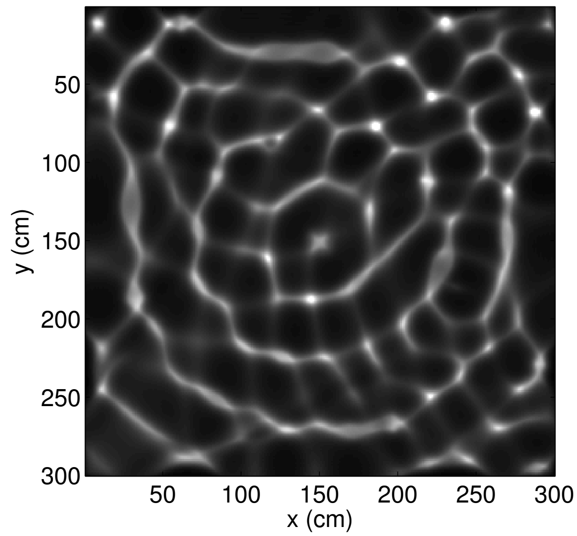


Figure 9: A result from the exploding surface model. [**CR**]

inversion becomes a combinatorial optimization problem which could be implemented using a heuristic search method. The right panel of Figure 10 shows one possible surface obtained using simulated annealing. Although the obtained solution might be physically inaccurate, the ray counts produced by the two surfaces in the figure are identical. This numerical non-uniqueness is the result of simplified implementation of the theory – finite ray tracing with binning.

Despite the large null space, a large number of solutions can be tested at a minimal cost because the full forward modeling is not needed. The desired solution lies within the null space. Most of the solutions in the null space are physically impossible. Therefore, these solutions can be discarded using an energy function. For example, for a pre-caustic slice the energy function can be the sum of distances traveled by the rays:

$$E(\text{solution}) = \sum_{i}^{N} |\vec{p}_i| \quad . \tag{9}$$

Minimizing the energy function proposed yields the correct solution if given enough iterations (see Figure 11). In fact, simulated annealing can be reduced to the greedy algorithm, which moves only toward better solutions. Inversion for intensity fields below caustics needs a more elaborate energy function. The resolution that binning provides is insufficient to resolve the caustics, especially where the caustics overlap. In three dimensions, there are more constraints on the solutions because neighboring quads of the surface must match for the surface to be continuous. Therefore, I expect that the search space reduces radically, and it can be traversed using a tree search algorithm to find a good approximate solution. Using this geometric search for a solution might not obtain the result but it can provide very cheaply a good approximation that can be used as a starting model for other inversion of real function.

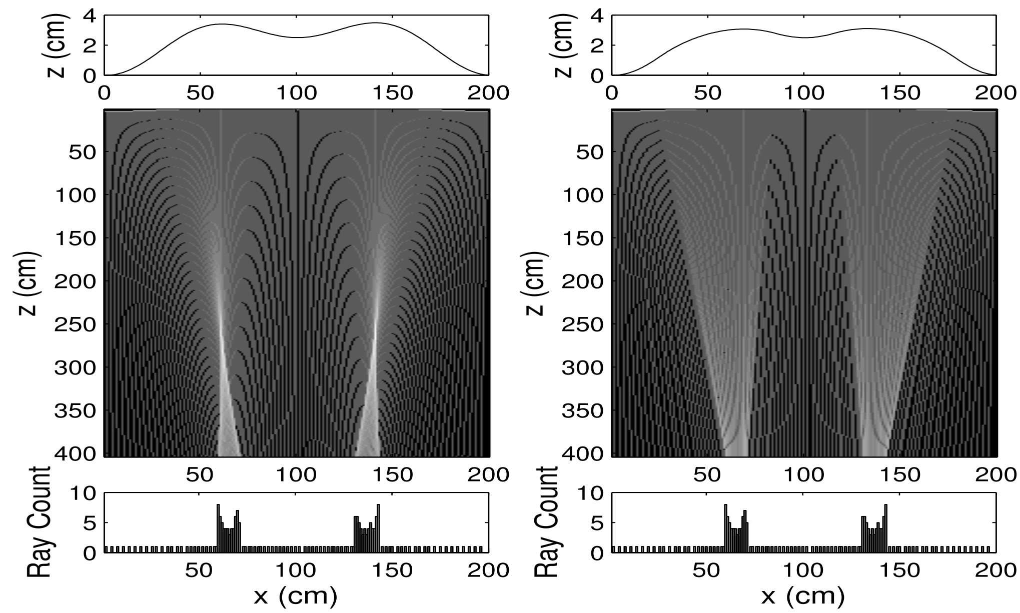


Figure 10: Left: ray tracing forward modeling for a surface. Right: One of the many surfaces obtained by inversion using ray-count. The upper graphs are the surface function, middle plots are the ray tracing of light under the surface, and the bottom plots show the ray count at the bottom of the pool. [**CR**]

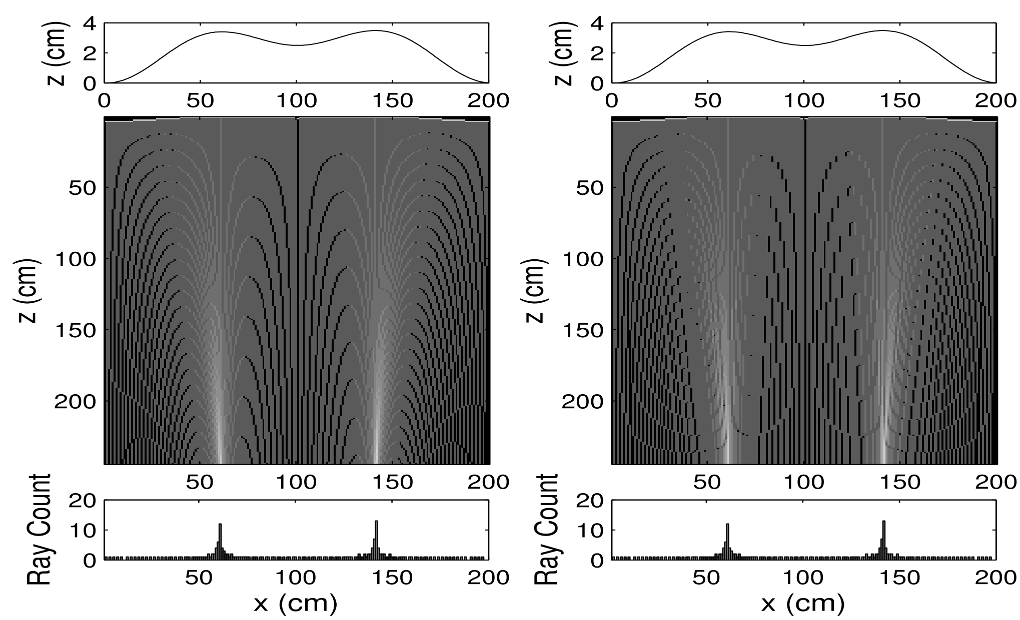


Figure 11: Left: ray tracing forward modeling of pre-caustics intensity field. Right: The inversion result by minimizing distances traveled by rays. [**CR**]

## CONCLUSIONS AND FUTURE WORK

In this paper, I show how we can construct the light intensity field at the pool floor using approximate operators. A light intensity field can provide information about the geometry of the water surface. Inversion using patterns obtained from ray tracing and binning suffers from a large null space– i.e. many surfaces can produce the same patterns at the bottom of the pool. Future directions include using the proposed geometric inversion approach in three dimensions, and formulating and implementing the inversion problem using a realistic light intensity field.

## ACKNOWLEDGMENTS

I thank Sjoerd de Ridder for sharing his ideas about stabilizing the forward modeling using distributions. Thanks to Jeff Shragge, Yaxun Tang, and the rest of SEP group for their contribution and for helping me during my struggle with the computing environment.

## REFERENCES

Chapra, S. and R. Canale, 2002, Numerical Methods for Engineers - Fourth Edition: The McGraw-Hill Companies Inc.

Claerbout, J. F., 1985, Imaging the earth's interior: Blackwell Scientific.

——, 2007, A tease for theoretical seismologists: WWW. (URL http://sepwww.stanford.edu/sep/jon/poolwaves).

de Ridder, S., 2008, Stable simulations of illumination patterns caused by focusing of sunlight by water waves: Technical report, SEP-134.

ICMM, P. C. G., 2008, Introduction to light diffusion: WWW. (URL http://luxrerum.icmm.csic.es/?q=node/research/light_diffusion).

Shah, M., J. Kontinnen, and S. Pattanaik, 2007, Caustics mapping: An image-space technique for real-time caustics: IEEE Trans. Vis. Comput. Graph, **13**, 272–280.

Watt, M., 1990, Light-water interaction using backward beam tracing: Computer Graphics, 377–385.

Yilmaz, O., 2001, Seismic Data Analysis: Society of Exploration Geophysicists.