**Introduction to Light**

In computer graphics it is essential to consider theoretical models from the physical world when rendering images. Our aim is to be as photorealistic as possible which requires us to compute and simulate real world physical phenomena as closely as possible.

One of the core theories we need to understand is the behaviour and properties of light. Lighting models are used in renderers to influence the final colour of the image pixels that correspond to certain objects in the scene. These objects have material properties that interact with the lighting to compute the final rendered colour of the pixel.

Beyond these models used in modern scanline rasterizers such as DirectX or OpenGL, we have another form of rendering known as ray tracing. Ray tracing is a technique of rendering that simulates the physical behaviour of light to produce photorealistic images. It has historically been slower than rasterizers although that has changed with modern GPU programming. Ray tracing will be discussed in more detail in other posts on this site.

So, to build an understanding of rendering we must first take the time to understand how light works in the real world and try and replicate that in our digital world through our calculations.

**Light and Energy**

Light is a form of energy that is transported as a stream of photons along a straight-line ray. The ray of light travels from the light source until it strikes a surface. The surface either absorbs or scatters the photons of light based on its material properties. The scattered photons bounce off the surface and the ray changes its direction. The process continues if and when the ray strikes further surfaces in the scene.

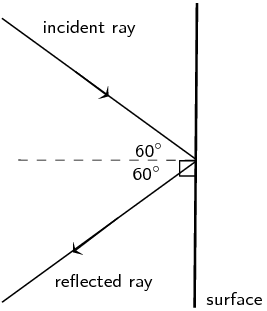


Figure 1 An example of elastic scattering where a ray strikes a surface, having a reflected angle equal to the angle of incidence

When a photon interacts with an atom you can have a number of behaviours:

* **Elastic scattering** where the photon keeps its energy (and phase) and changes its angle. An example of this is a mirror reflection.
* **Inelastic scattering** where the photon transfers some of its energy to the atom and changes angle. Here you can imagine an infrared light transferring some of its kinetic energy to the vibrational motion of molecules of an object thereby heating it up.
* **Absorption** where the photon transfers all of its energy to the atom.

Beyond this change in the energy of a photon in a ray, the overall energy and frequency of the scattered ray is impacted by whether the surface it impinges upon is moving and the direction it is moving. If the surface is moving towards the ray of light, energy is transferred from the surface to the scattered light ray increasing the energy and frequency of the ray. If the surface is moving away from the ray of light that reverse occurs, and energy is transferred from the ray to the surface and the overall energy and frequency of the ray decreases. If the surface has equal and opposite momentum to the ray, the overall energy and frequency of the ray remains unchanged beyond any changes of energy from the interaction of the rays’ photons with atoms of the surface.

This is a principle used by the Doppler Radar, which is used in technology such as speeding guns where microwaves are bounced off a surface to determine its speed.

In order to realistically simulate light in our digital world we use radiometry, which is the science of measuring the physical properties of light such as wavelength, energy and power.

In order to measure the light in our digital world we need to develop a quantitative model of light. To see how we would use this measure of light, consider how real-world images are generated. They are formed in cameras by measuring the amount of light that enters the cameras aperture and then strikes each pixel on the cameras image sensor. We do the same thing when rendering. We want to measure the total amount of light that strikes each pixel on our virtual cameras’ image sensor.

**Light Rays**

We have already discussed that light travels as a stream of photons that travel along a ray. We need a way to mathematically model that ray.

Any point **P** on a ray can be given by the following equation:

**P = P0 + P1t**

Where **P0** is the origin of the ray. In other words, where the ray of light is being emitted from or has been scattered from.

**P1** is the direction the ray of light is travelling.

**t** is the distance the ray has travelled.

**P**, **P0** and **P1** are all 3D vectors where P and P0 are position vectors representing a point in space.

This equation allows us to model the stream of photons as a finite line rather than an infinite ray and we can solve mathematically for the value of t to find the point on the ray at an intersection with a given object in the scene. We will look at ray-object intersection in later posts.

The transport path of the light can be considered as a collection of line segments that are represented by a starting point and ending point. When we know the starting point we have an orientation for the line segment which is important for determining the direction light is travelling. A surface that emits photons can be interchangeably referred to as an emitter in the general case, luminaire in the case of a non-natural light source such as a light bulb or a light source when relatively small in extent.

**Light Fields**

To understand how our simulated light rays interact with our scene we need a understand theoretical model known as a light field.

Light fields are commonly used in a variety of domains such as astronomy, art and optics. The term was coined by A. Gershun in his 1936 paper titled ‘The Light Field’ and the idea of a field of light can be traced back as far as the 15th Century in Da Vincis’ manuscripts.

The light field consists of the sum of light in a 3-dimensional space that is flowing in every direction through every point in that space. A light field can be considered in both 5 dimensions and 4 dimensions.

The 5-dimensional light field is given by the plenoptic function. The plenoptic function is used in computer graphics to express the image of a scene from any possible viewing position at any point in time. It was described as the radiance of all rays in a region of 3-dimensional space illuminated by an unchanging arrangement of lights by Adelson in his 1991 paper ‘The plenoptic function and the elements of early vision’. The radiance is the measure of the amount of light in the ray and is given by the magnitude of the ray.The 5-dimensional ray is expressed in terms of three coordinates, **x**, **y** and **z** and two angles **θ** and **ϕ**. The ray is depicted below passing through point (x, y, z) with direction (θ, ϕ) where the radiance is denoted by L.

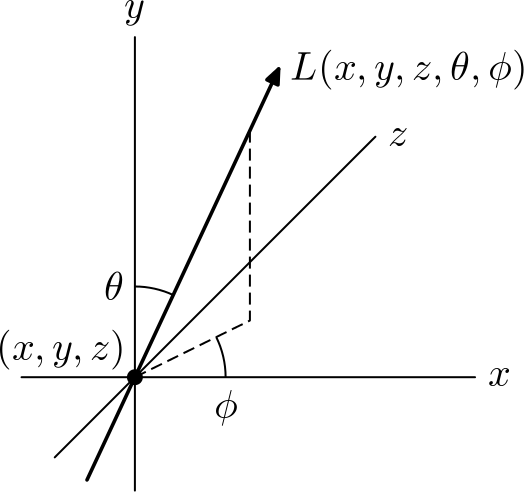


Figure 2 A 5D of position (x, y, z) and direction(θ, ϕ) ray depicted in a 3D space

Gershun similarly regarded the light field as a 5D function, however, he regarded it as an infinite collection of vectors, one for every direction striking a point and where the radiance of the vector is equal to its magnitude. By integrating these vectors in every direction, or more simply, in the direction of every light in a collection, from any point in the scene the result is a scalar and a direction. The scalar is the total irradiance or output of light energy for the given point and the direction is the direction of that resulting irradiance. This direction is the direction you would face a flat surface at that point in order to most brightly illuminate it.

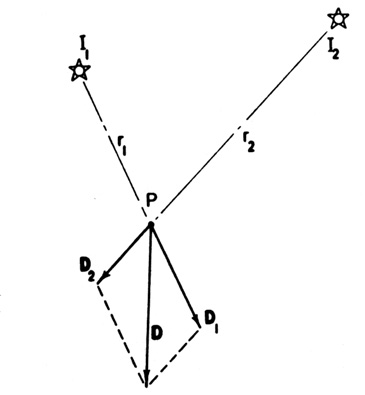


Figure 3 Gershuns' light field solution for the irradiance of point P summing the irradiances D1 and D2 from light sources I1 and I2 to produce the resultant irradiance vector D

In computer graphics this theory is known as the vector irradiance field as coined by Arvo in 1994.

The 4D light field ‘completely characterizes the flow of light through unobstructed space in a static scene with fixed illumination’ (M Levoy).

In a 4D light field a ray has constant radiance from point to point along its length. Given the radiance is constant we can drop a coordinate from the ray as we only need to know a single point on the ray to calculate its radiance. This leaves us with a 4-dimensional parameterization for the ray.

We will consider the 5D light field for the rest of our discussions.

**Light Transport Paths and Conservation of Light**

Here we assume that the space in our scene between objects is ‘empty space’. In other words it is a vacuum or air like space. We also assume that there is no absorption of light, which we would have in a real-life scenario.

In this ‘empty space’ environment all energy is conserved along the ray; no new light joins the ray and no light leaves the ray. This means that all the light leaving a point **X** in the directionof another point **Y** must all reach that point **Y** and vice versa.

This is very useful for us in computer graphics to compute efficient rendering of images. Rather than having to compute lighting calculations for the infinite number of points along the path of the ray, we simply have to consider the points at which the ray of light changes direction (or in other words collisions with object in our scene).

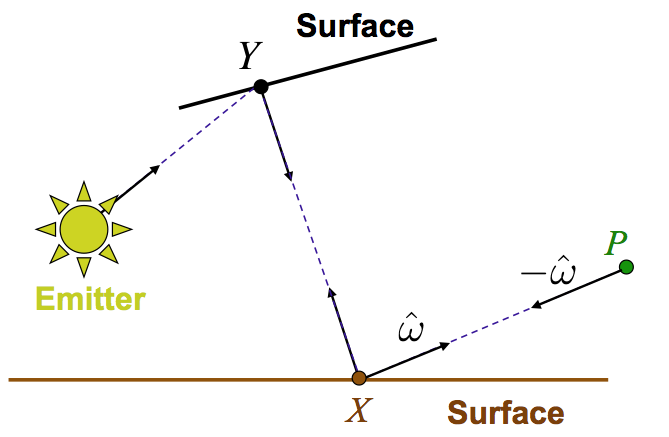


Figure 4 Light transport path from emitter to point P. The light ray scatters from point Y to point X and finally to point P

We can express the conservation light along a ray as:

**L(P – εŵ, ŵ) = L(X + εŵ, ŵ)**

Where **ŵ** is the normalized direction from point **X** to point **P**.

We use **ε** as a minute constant amount by which to multiply the direction vector **ŵ**. This allows us to consider a point slightly offset from the point of the surface, between the two points, rather than looking directly on the surface as what happens here can be ambiguous. This is because the exact point on the surface may be completely opaque to light allow no light to pass through it. In this case, the intensity of light at this point is 0.

We can further simplify the light intensity expression for a point by considering the incoming and outgoing light from that point instead of using **ε** to offset from that point.

We can express the outgoing light at point **X** as:

**Lo(X, ŵ) = L(X+ εŵ, ŵ)**

The incoming light can be written as:

**Li(X, ŵ) = L(X - εŵ, -ŵ)**

This makes it much easier to consider a ray of light along its entire path, accounting for its changes in direction from collisions, rather than just looking at individual light segments between two points.

We can now rewrite the conservation of light equation as:

**Li(P, ŵi,p) = Lo(X, ŵo,x)**

We can now use this equation to calculate the amount of light striking a pixel in our camera image sensor. This calculation will allow us to use the light striking the camera to compute the colour of each pixel and is thus essential to rendering an image.

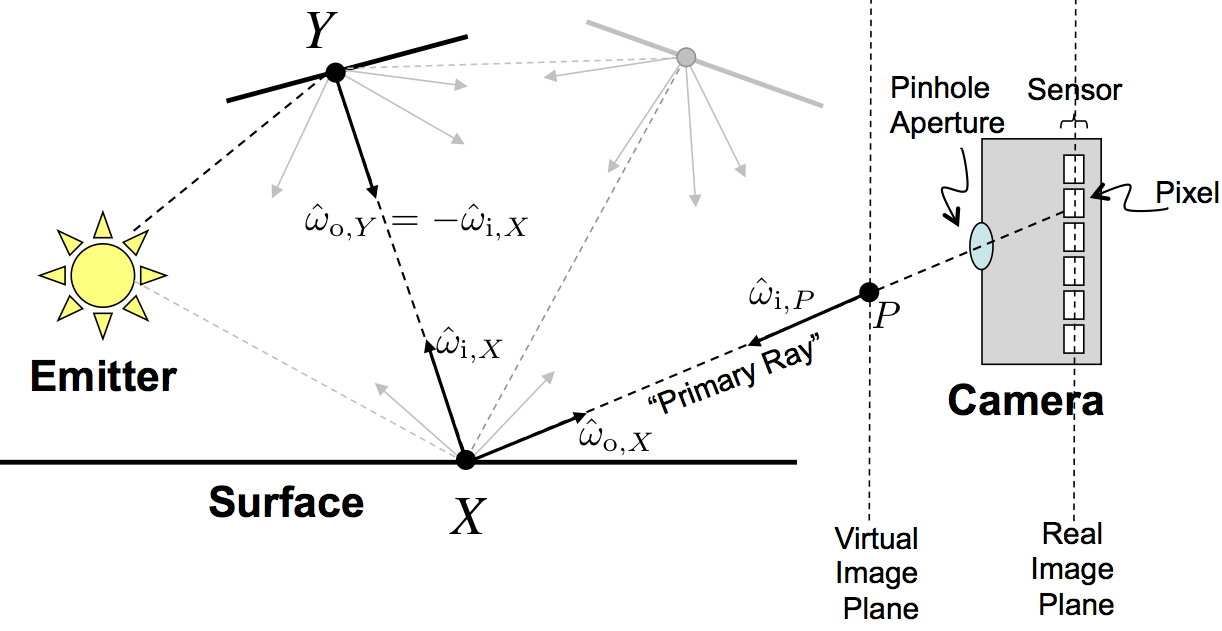


Figure 5 A complete light transport path from the emitter to the image sensor in our virtual camera, striking points Y and X along its path

To consider the light leaving **X** we have to consider light entering **X** from **Y**. The light leaving **X** for the given ray will be equal to the light scattered from **X** that was received from **Y**  plus any light that was emitted from **X** such as in the case that **X** is glowing. The emitted light will be very simple to implement in a renderer and is simply computed using the emissive material property of the surface.

Given the light field consists of every ray of light flowing in every direction through every point, we have to consider the light striking **X**  from every direction, not just from **Y**.

To consider the scattered light reaching **P** from **X**, we need to consider every ray of light that has **XP** as its final ray segment. To do that we have to consider all the points from which light may reach **X**.

We do that with the following equation:

**Lo(P, ŵo,x) = Le(X, ŵo,x) +** ∫visible points **Li(X, ŵi,x)** ...□ ... dY

Here we integrate over every point **Y** visible from **X**.

**Le** represents the light emitted from **X**.

□ represents the light scattered from **X** as well as some measurement details about the integration which we will elaborate on further in the next part discussing the rendering equation.

It is also worth noting we have made a number of assumptions up to this point. So far we haven’t given any units for light so we are unable to measure it. We will discuss the measurement of light in the next unit and use it to complete the equation.

So far, we have only considered steady state light transport in a specific moment in time. This can have an impact in real time rendering as object positions and lighting conditions can change over the course of a frame so if we wanted to render phenomena such as motion blur we would need to analytically or numerically integrate over time.

We also do not consider the frequency of the light as it corresponds to the red, green and blue cones in the human eye. In the real-world certain materials will appear differently under different frequencies of light.