Combing Hair and Sampling Directions

Jeppe Revall Frisvad

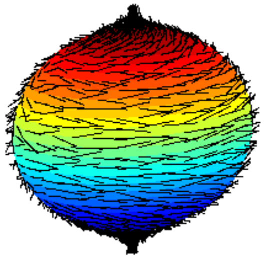
December 2014 (extra slide added October 2018)

The Hairy Ball Theorem

- An even-dimensional sphere does not possess any continuously differentiable field of unit tangent vectors.

- Is it possible to comb a coconut flat?

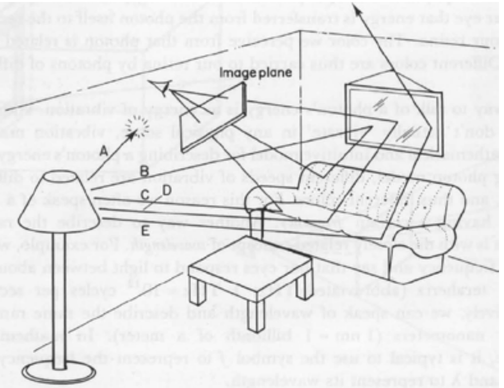
- How do we comb it well?

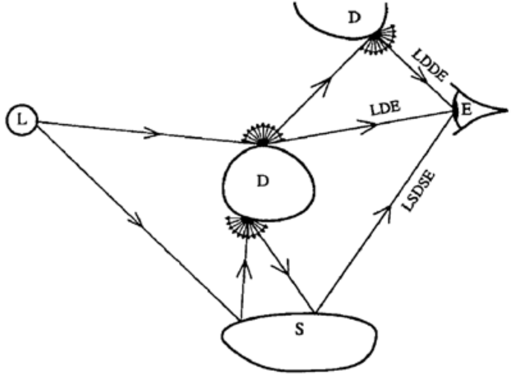


Failed attempt to comb a hairy ball,
leaving an uncomfortable tuft at each pole [Wikipedia]

- Is it possible to consistently pick a perpendicular vector?

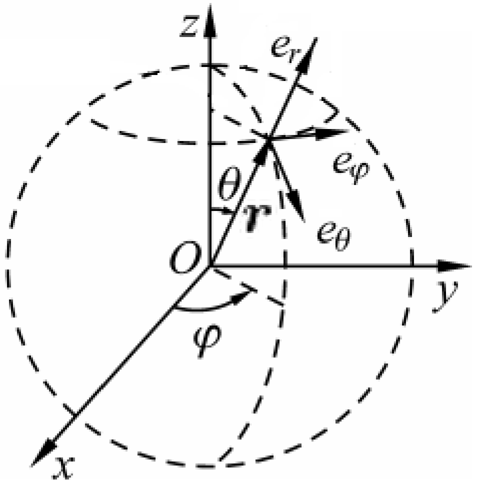
Monte Carlo Ray Tracing



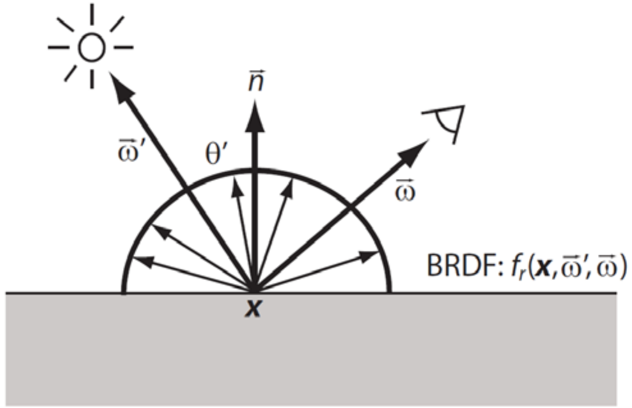


- One of the most commonly used operations in MC ray tracing is to sample a direction according to a distribution around a normal.
  - *- Directions will be sampled millions and millions of times in a common rendering*.

- The direction is usually sampled in spherical coordinates.

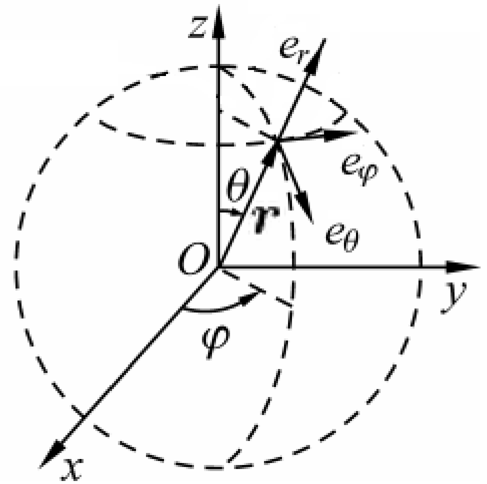


- We need a change of basis.



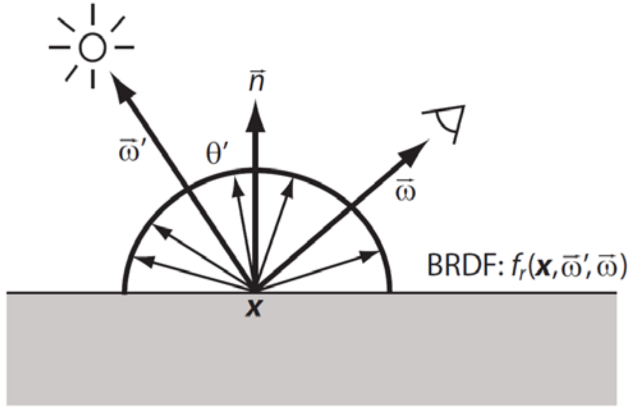
- How do we efficiently get this rotation?

Sample Space to World Space



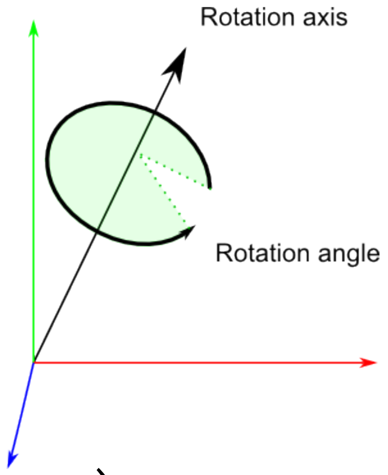
- We sample spherical coordinates (θ, ϕ).
- We convert to a local coordinate system with basis $\left\{\vec{b}_1, \vec{b}_2, \vec{n}\right\}$ by
$$(x, y, z) = (\sin\theta\cos\phi\,, \sin\theta\sin\phi\,, \cos\theta).$$

- The change of basis is simply
$$\vec{\omega} = x\,\vec{b}_1 + y\,\vec{b}_2 + z\,\vec{n}.$$
but we typically have no explicit knowledge of $\vec{b}_1$ and $\vec{b}_2$.



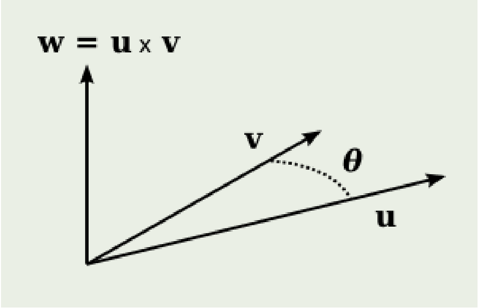
- We need a technique for building an orthonormal basis from a unit vector.

Rotation from One Vector to Another



- If we use quaternions, we do not need a basis.
- The unit quaternion $\hat{\boldsymbol{q}}$ that specifies the rotation from $\boldsymbol{u}$ to $\boldsymbol{v}$:

$$\hat{\boldsymbol{q}} = (\boldsymbol{q}_v, q_w) = \left(\sin\frac{\theta}{2}\boldsymbol{w}, \cos\frac{\theta}{2}\right) = \left(\frac{\boldsymbol{u}\times\boldsymbol{v}}{\sqrt{2(1+\boldsymbol{u}\cdot\boldsymbol{v})}}, \frac{1}{2}\sqrt{2(1+\boldsymbol{u}\cdot\boldsymbol{v})}\right).$$



- Let us set $\boldsymbol{u} = (0,0,1)$ and $\boldsymbol{v} = \vec{n} = (n_x, n_y, n_z)$, then

$$\hat{\boldsymbol{q}} = \left(\frac{(-n_y, n_x, 0)}{\sqrt{2(1+n_z)}}, \frac{1}{2}\sqrt{2(1+n_z)}\right).$$

- Applying $\hat{\boldsymbol{q}}$ to our sampled direction, we rotate to world space: $\hat{\boldsymbol{q}}\,(x,y,z,0)\,\hat{\boldsymbol{q}}^*$.

Quaternion Versus Change of Basis Matrix

- Quaternion:
  - No need to build a basis.
  - Involves square roots and quaternion multiplications.
  - Singularity for $\vec{n} = (0, 0, -1)$.

- Change of basis matrix:
  - Build an orthonormal basis from the normal.
  - Involves cross products and vector normalization.
  - Avoid choosing a vector parallel with the normal when building the basis.

- Both methods accomplish the same.
- What basis is created by the quaternion technique?

Building an Orthonormal Basis [Frisvad 2012]

- Doing some lengthy calculations and simplifications, I found that

$$(\hat{\boldsymbol{q}}\,(x, y, z, 0)\,\hat{\boldsymbol{q}}^*)_v = \left(x \begin{pmatrix} 1 - n_x^2/(1+n_z) \\ -n_x n_y/(1+n_z) \\ -n_x \end{pmatrix} + y \begin{pmatrix} -n_x n_y/(1+n_z) \\ 1 - n_y^2/(1+n_z) \\ -n_y \end{pmatrix} + z \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix} \right).$$

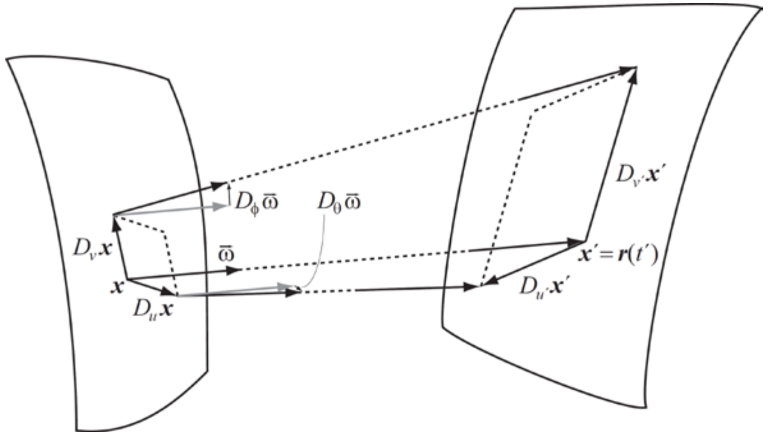
- This is an extremely efficient way to rotate sampled directions to world space (no square roots, easy to optimize).
- As a by-product, it is also a new way to build an orthonormal basis from a 3D unit vector:

$$\vec{b}_1 = \left(1 - \frac{n_x^2}{1+n_z}, -\frac{n_x n_y}{1+n_z}, -n_x\right), \qquad \vec{b}_2 = \left(-\frac{n_x n_y}{1+n_z}, 1 - \frac{n_y^2}{1+n_z}, -n_y\right).$$

Rendering Advantages

- Faster sampling of directions
  (sampling of spherical coordinates becomes advantageous compared to von Neumann rejection sampling.)

- We can use it to produce tangent vector directions that are consistent (no discontinuities).



- Thus, this method is probably also good for combing of hairy objects.

[demo]

Avoiding the singularity and the branching

- Duff et al. [2017] improved the sampling technique by flipping signs in the basis so that we avoid numerical issues around the singularity.

$$\vec{b}_1 = \left(1 - \frac{n_x^2}{1+|n_z|}, -\frac{n_x n_y}{1+|n_z|}, -n_x \operatorname{sgn}(n_z)\right), \qquad \vec{b}_2 = \left(-\frac{n_x n_y}{1+|n_z|} \operatorname{sgn}(n_z), \left(1 - \frac{n_y^2}{1+|n_z|}\right) \operatorname{sgn}(n_z), -n_y\right).$$

- The trade-off is that the tangent vector directions are no longer consistent across the equator of the unit sphere.

```
inline void rotate_to_normal(const optix::float3& normal, optix::float3& v)
{
  const float sign = copysignf(1.0f, normal.z);
  const float a = -1.0f/(1.0f + fabsf(normal.z));
  const float b = normal.x*normal.y*a;
  v =   optix::make_float3(1.0f + normal.x*normal.x*a, b, -sign*normal.x)*v.x
      + optix::make_float3(sign*b, sign*(1.0f + normal.y*normal.y*a), -normal.y)*v.y
      + normal*v.z;
}
```