Photorealistic Motion

Fabian Andres Prada Niño

IMPA

Part I: Motivation

Part I: Motivation

- Explore the relation between time and photography.

Part I: Motivation

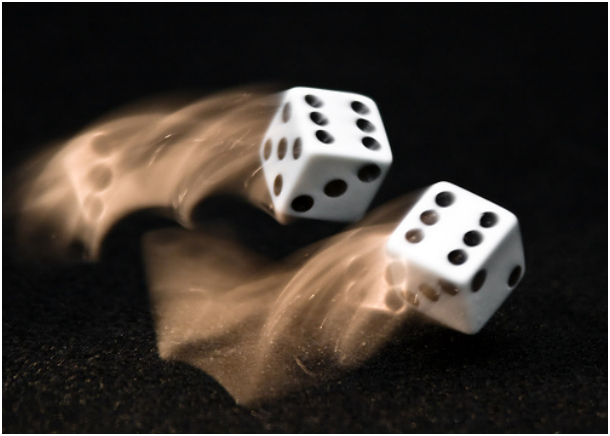
- Explore the relation between time and photography.
- Identify the advantages provided by Ray Tracing to simulate motion photography

Part I: Motivation

- Explore the relation between time and photography.
- Identify the advantages provided by Ray Tracing to simulate motion photography
- Explore the tools provided by PBRT to construct and render dynamic scenes

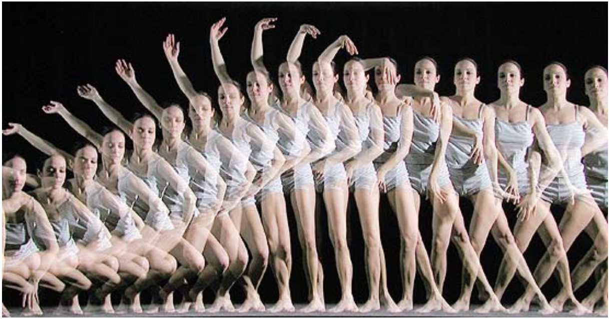
Motion in Photography: Motion Blur

Motion in Photography: Motion Blur



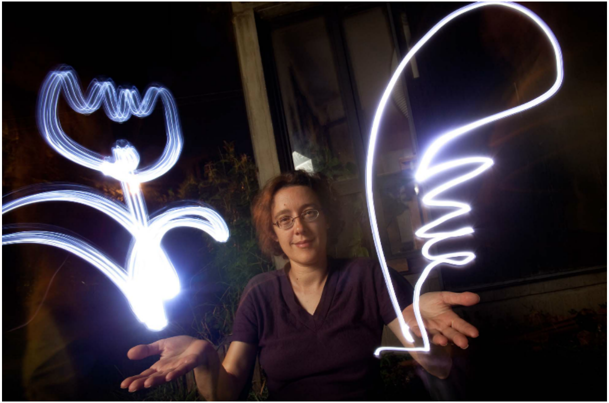
Motion in Photography: Strobe Photography

Motion in Photography: Strobe Photography



Motion in Photography: Light Painting

Motion in Photography: Light Painting



Motion and Ray Tracing. DRT[1]

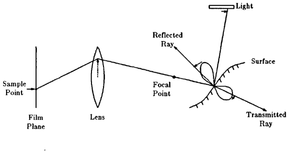
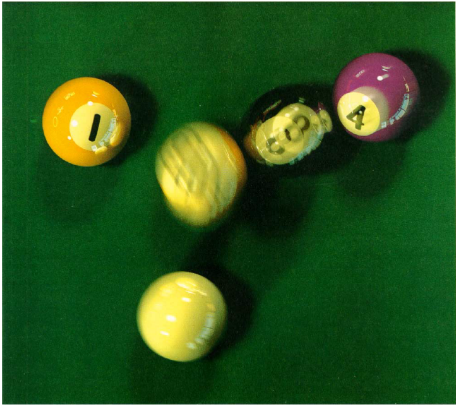
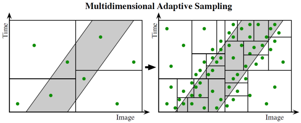
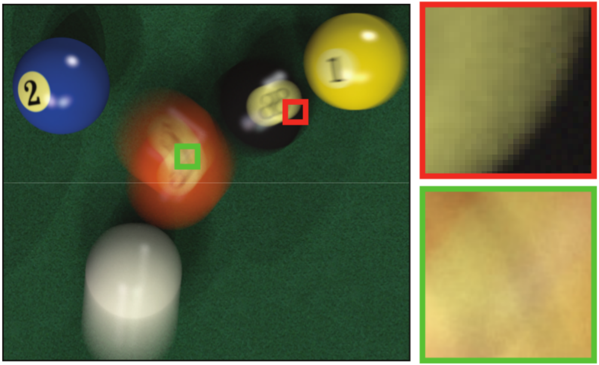


Figure 2. Typical Distributed Ray Path



Motion and Ray Tracing. MAS[1]





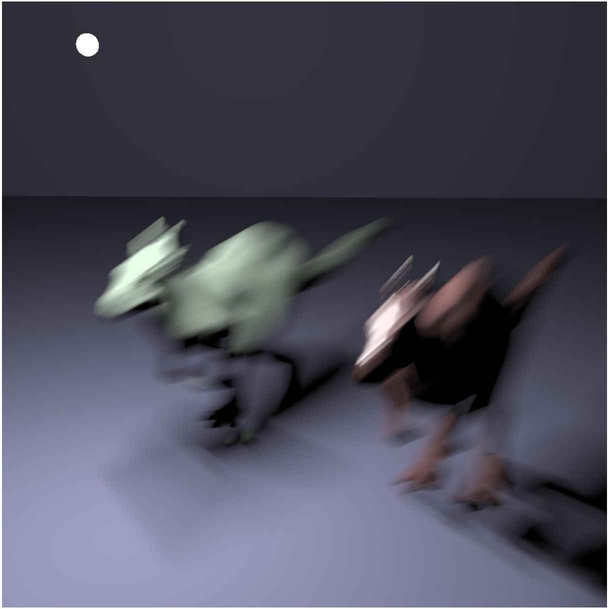
PBRT Features

TransformedPrimitive → AnimatedTransform

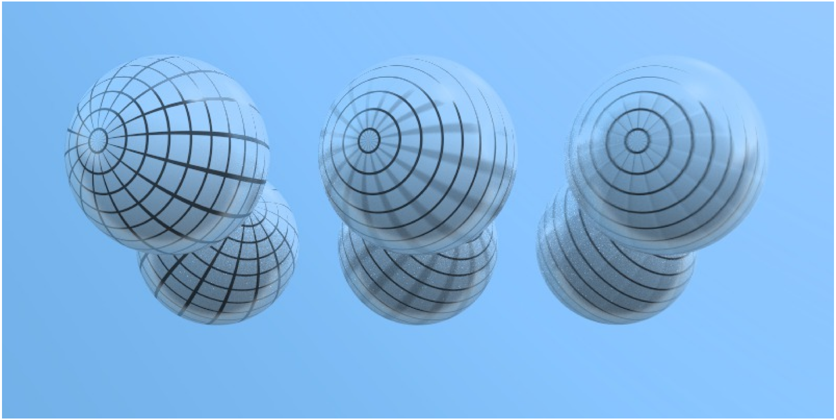
$$\text{AnimatedTransform} \rightarrow \begin{cases} \text{StartTransform} & \rightarrow M_0 = S_0 R_0 T_0 \\ \text{EndTransform} & \rightarrow M_1 = S_1 R_1 T_1 \end{cases}$$

$$\begin{aligned} \text{Transform(t)} &= \text{Interpolate(StartTransform,EndTransform,t)} \\ &= \text{Interp}(S_0, S_1, t) * \text{Interp}(R_0, R_1, t) * \text{Interp}(T_0, T_1, t) \end{aligned}$$

PBRT Example 1



PBRT Example 2



Part II: Implementation

Part II: Implementation

- Events handle: Flashing and Instants Description.

Part II: Implementation

- Events handle: Flashing and Instants Description.
- Extension of the motion transformations provided by PBRT.

Part II: Implementation

- Events handle: Flashing and Instants Description.
- Extension of the motion transformations provided by PBRT.
- Time dependent attributes for light: Intensity and Position.

New Classes

New Classes

- LightEvents $\begin{cases}\text{InstantsDescriptor}\\ \text{DynamicSamplerRenderer}\end{cases}$

New Classes

- LightEvents $\left\{\begin{array}{l}\text{InstantsDescriptor}\\ \text{DynamicSamplerRenderer}\end{array}\right.$
- DynamicTransforms $\left\{\begin{array}{l}\text{DynamicPrimitives}\\ \text{DynamicLights}\end{array}\right.$

New Classes

- LightEvents $\begin{cases}\text{InstantsDescriptor}\\ \text{DynamicSamplerRenderer}\end{cases}$
- DynamicTransforms $\begin{cases}\text{DynamicPrimitives}\\ \text{DynamicLights}\end{cases}$
- LightIntensity

LightEvents

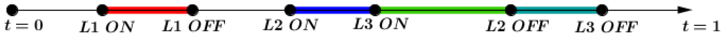
LightEvents : API

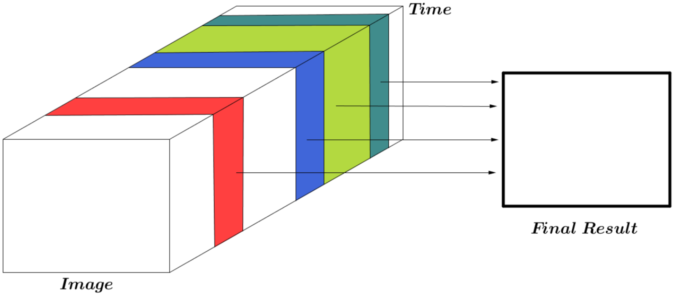
World Begin

LightSource ”spot” ”float tInitial”[0.3] ”float tFinal”[0.6]

World End

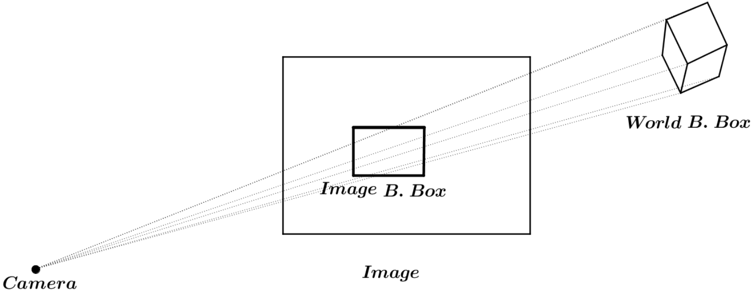
LightEvents Renderization



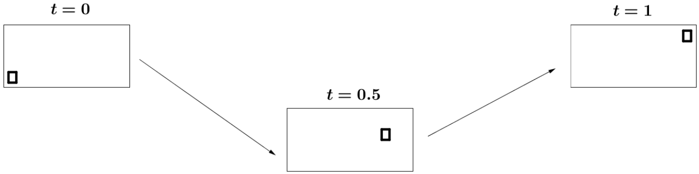


LightEvent Refinement: Instant Descriptor

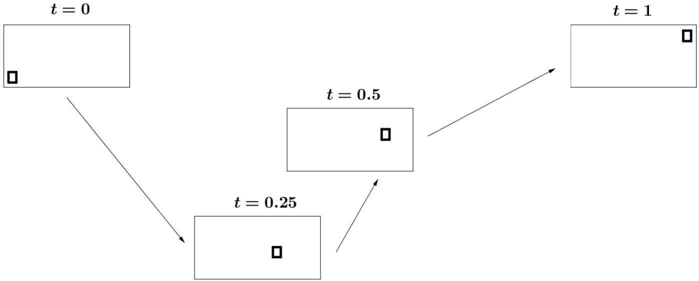
Instant Descriptor: (Time, Image Bounding Box)



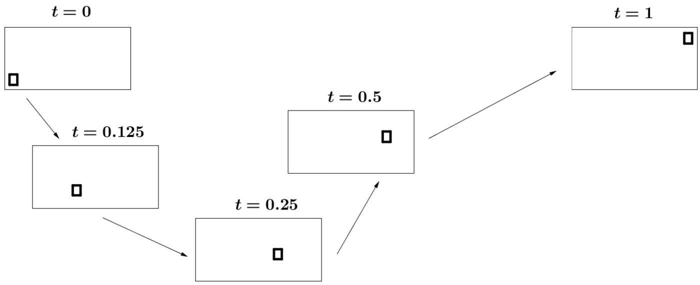
LightEvent Refinement: Subdivision by Image B. Box Movement



LightEvent Refinement: Subdivision by Image B. Box Movement



LightEvent Refinement: Subdivision by Image B. Box Movement



DynamicTransforms

DynamicTransforms : API

```
World Begin

DynamicTransform ”parabolic” ”point initialPosition” [0 1 0]
”vector initialVelocity” [3 0 0] ”vector acceleration” [0 0 10]
”vector axis” [1 1 0] ”float cycles” [5]

AttributeBegin
SetDynamicOn
Shape ”sphere”
SetDynamicOff
AttributeEnd

World End
```

DynamicTransform : Classes Structure

```
class DynamicPrimitive : public Primitive{
⋮
DynamicTransform * dynamics;
}

class Transform{
⋮
friend DynamicTransform;
}

virtual class DynamicTransform{
⋮
virtual void setTransformationMatrix(float time, Transform * t);
}
```

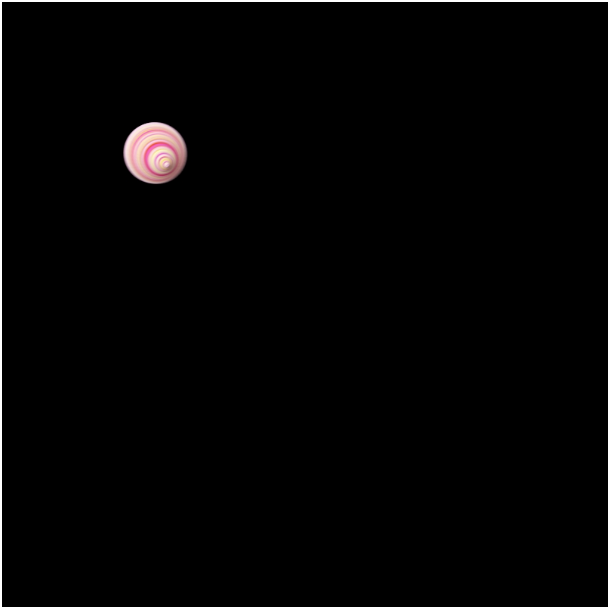
DynamicTransform : Classes Structure

```
class parabolicMovement: public DynamicTransform{
public:
⋮
void setTransformationMatrix(float time, Transform * t);
⋮
private:
Point initialPosition;
Vector initialVelocity;
Vector acceleration;
Vector axis;
float cycles;
}
```

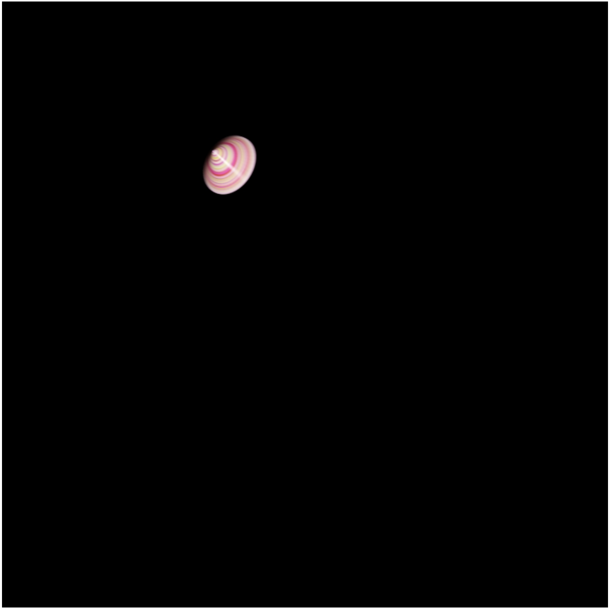
DynamicTransforms : Types

- Parabolic
- Piecewise Linear
- Helicoidal
- Simple Rotational

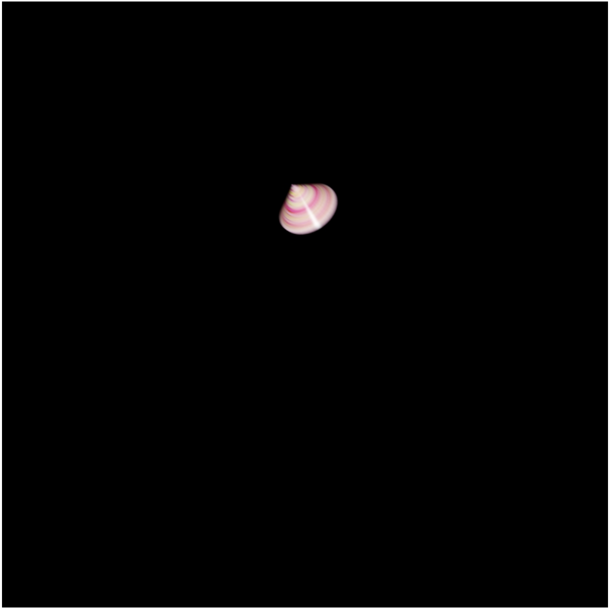
Parabolic



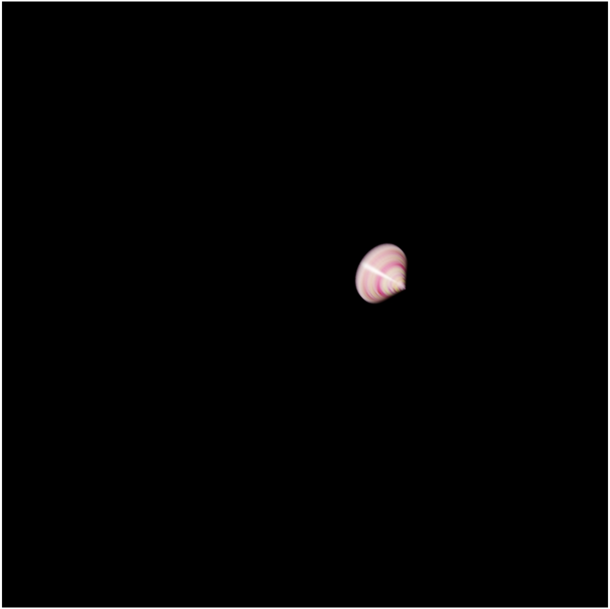
Parabolic



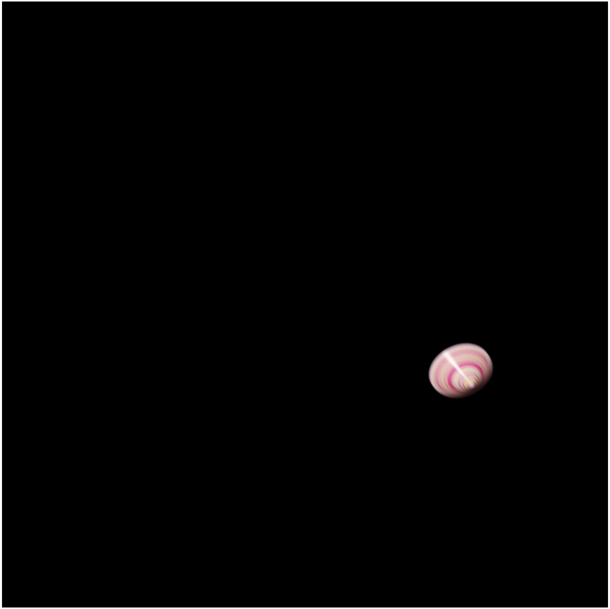
Parabolic



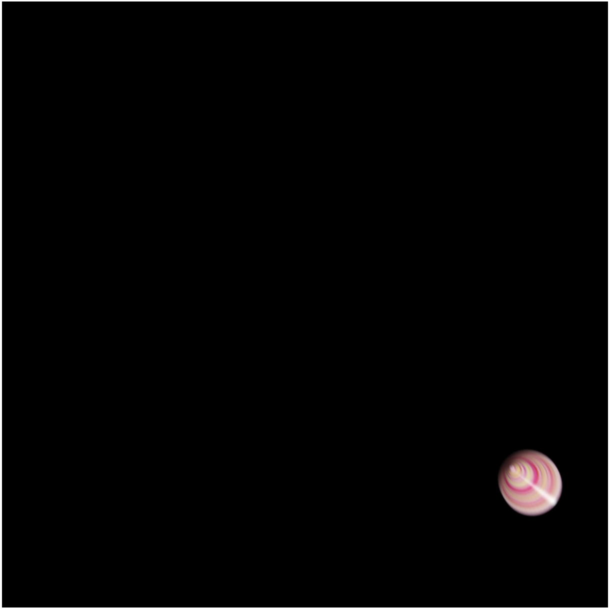
Parabolic



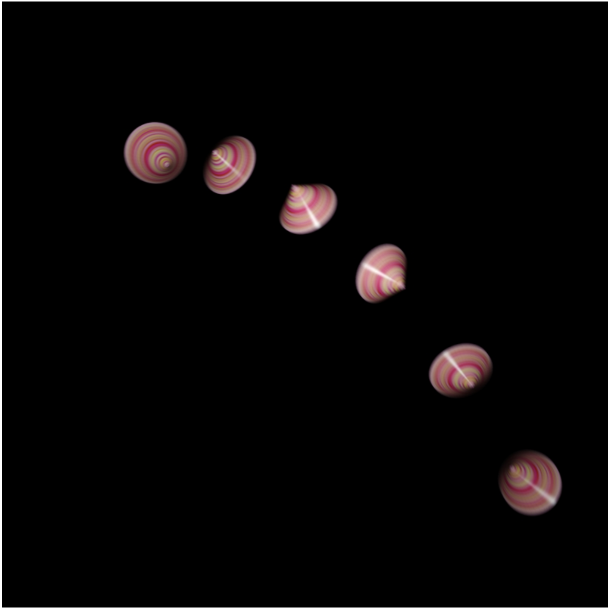
Parabolic



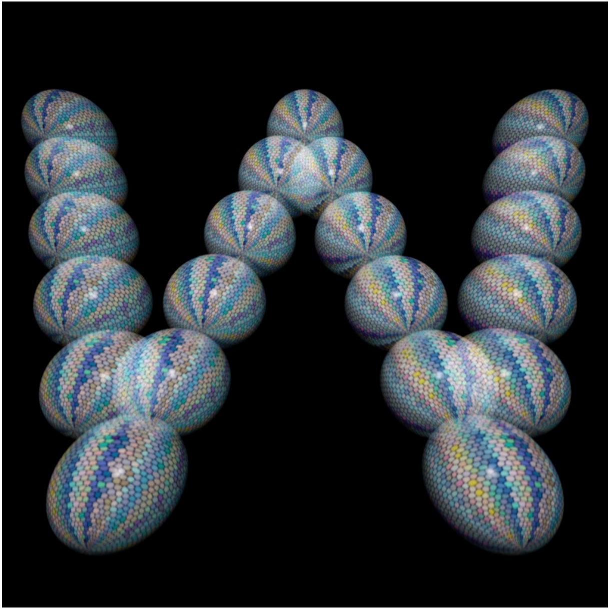
Parabolic



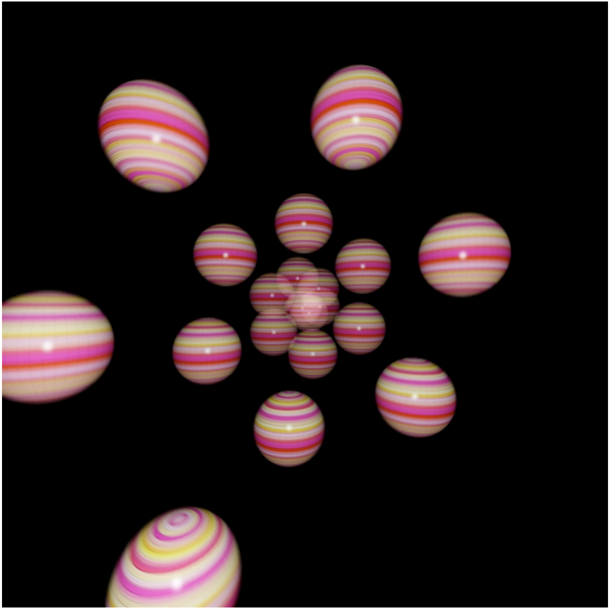
Parabolic



Piecewise Linear



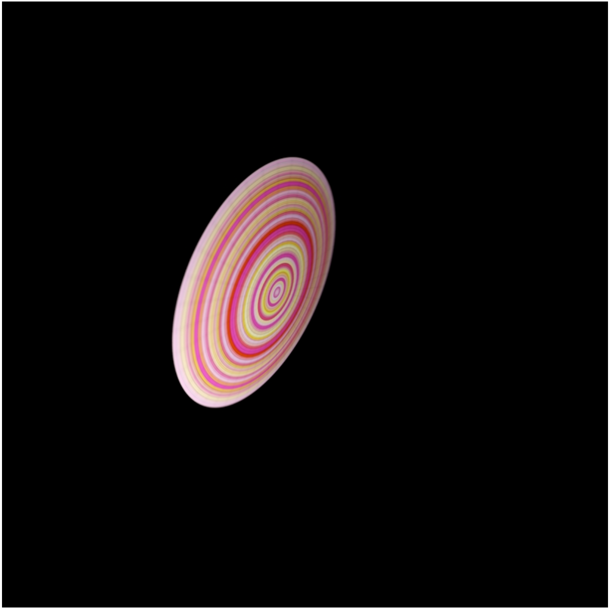
Helicoidal



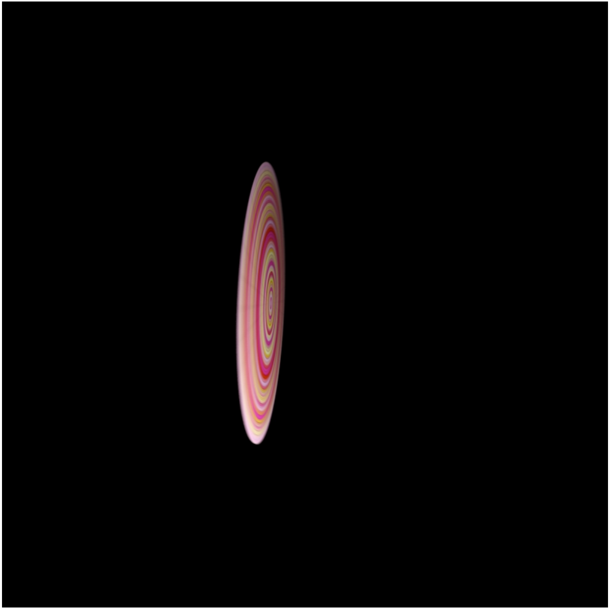
Simple Rotational

Simple Rotational

Simple Rotational



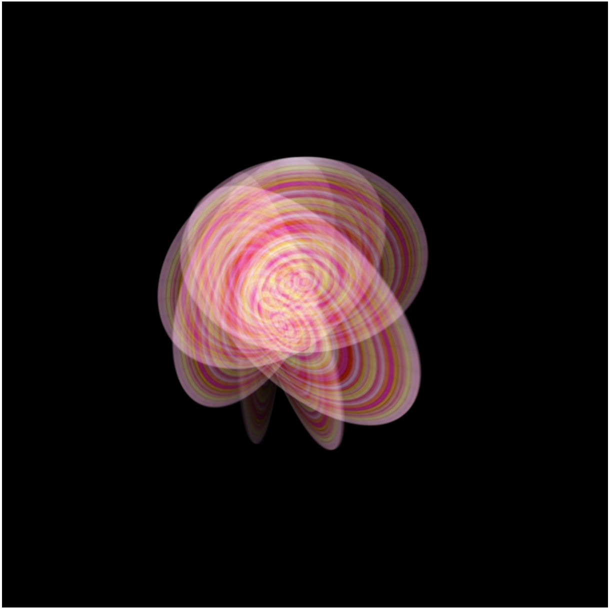
Simple Rotational



Simple Rotational

Simple Rotational

Simple Rotational



Dynamic Lights

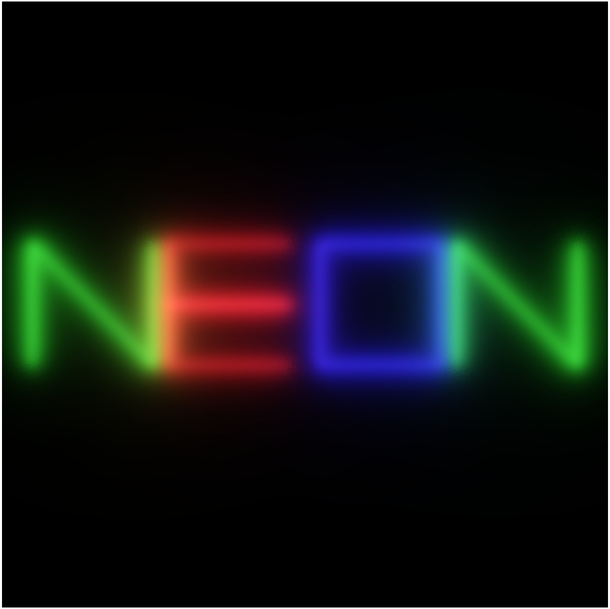
Dynamic Lights: Class Structure

```
class dynamicPointLight:Light{
⋮
DynamicTransform * dynamics;
}

class dynamicSpotLight:Light{
⋮
DynamicTransform * dynamics;
}
```

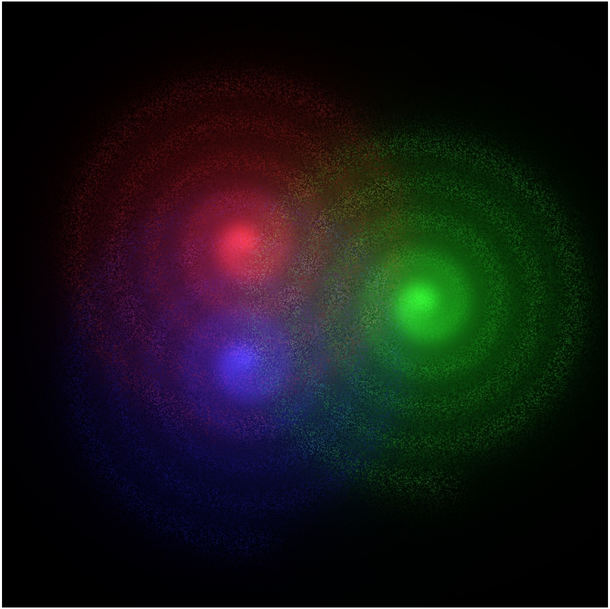
Dynamic Lights: API

```
World Begin

DynamicTransform ”piecewiseLinear” ”float timming” [0 0.5 1]
”float positions” [1 0 0 1 1 0 1 1 1]

SetDynamicOn
LightSource ”dynamicSpot”
SetDynamicOff

World End
```

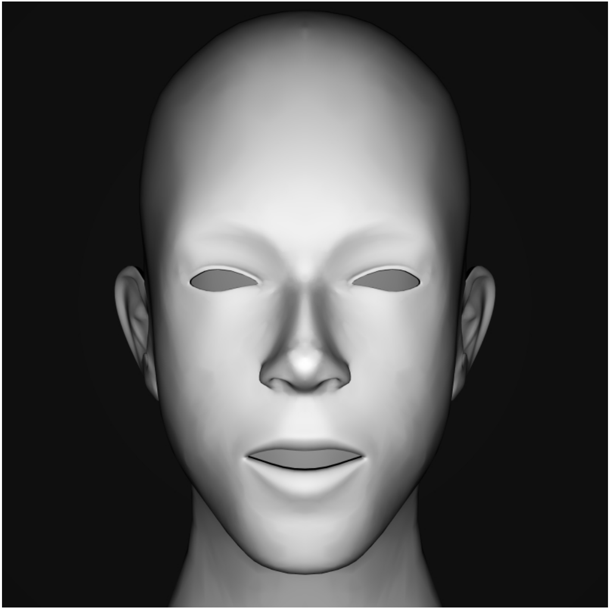
Dynamic Point Light: Example 1



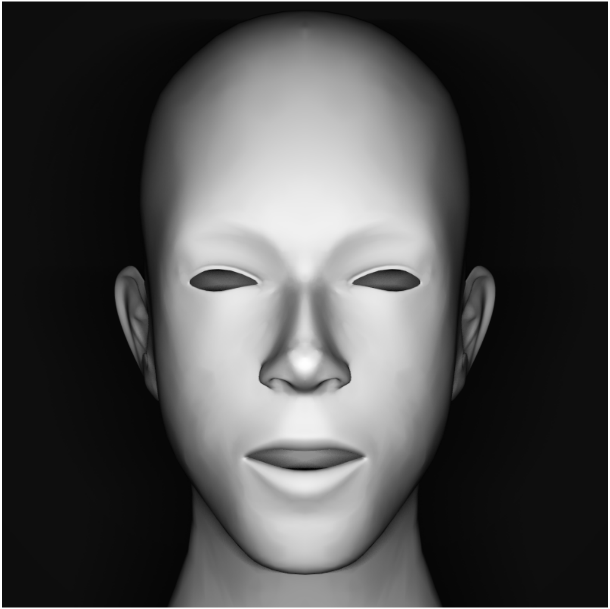
Dynamic Point Light: Example 1



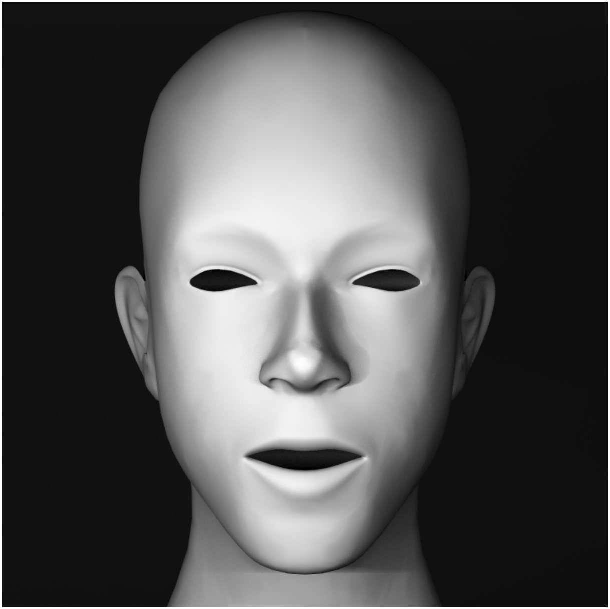
Spot Light. Hard Shadow



Dynamic Spot Light. Soft Shadows : Linear Movement

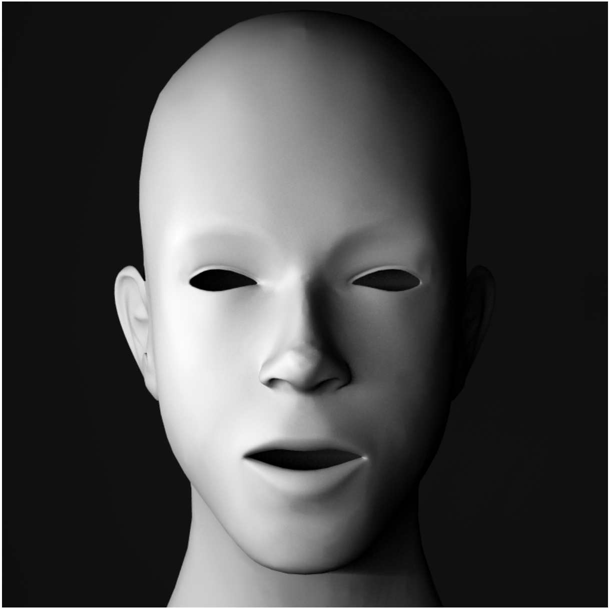


Dynamic Spot Light. Soft Shadows : Accelerated Mov.



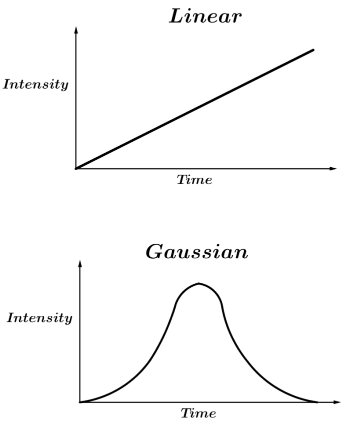
Dynamic Spot Light. Soft Shadows : Complete Rotation

Dynamic Spot Light. Soft Shadows : Quarter Rotation



LightIntensity

LightIntensity: Types



Linear

Hat

Gaussian

Exponential

LightIntensity : API

World Begin

LightIntensity ”gaussian” ”float mean” [0.5] ”float deviation” [0.1]

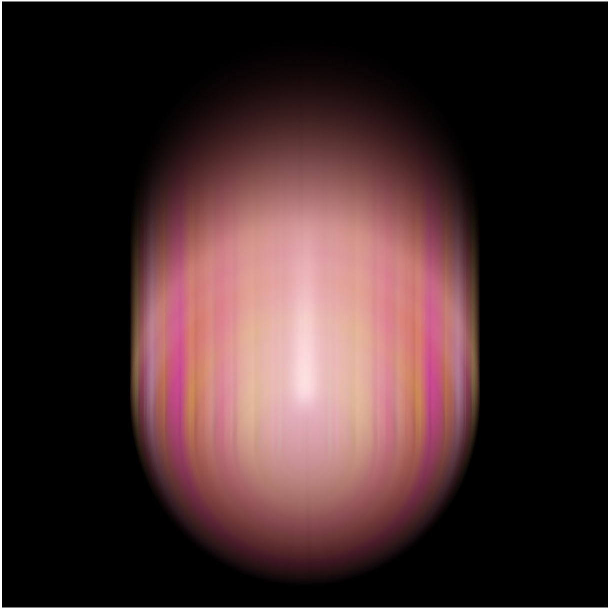
LightSource ”spot”

World End

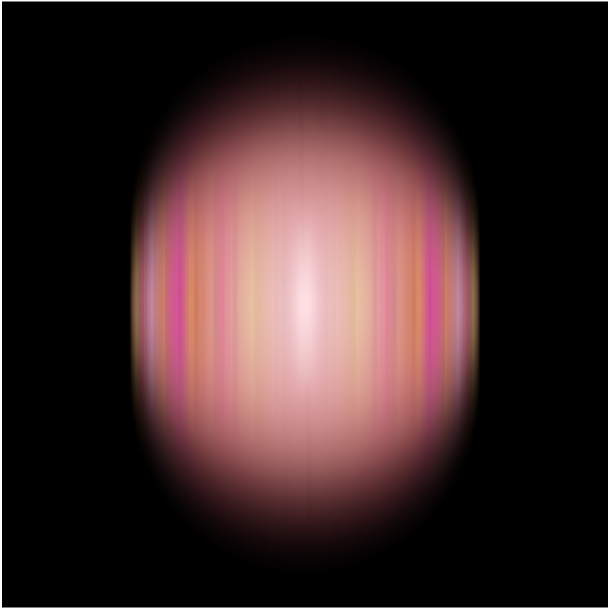
LightIntensity : Class Structure

```
class SpothLight: public Light{
⋮
LightIntensity * intensity;
}


virtual class LightIntensity{
virtual float intensity(float time);
}


class gaussianIntensity: public LightIntensity{
public:
float intensity(float time);
private:
float mean;
float deviation;
}
```

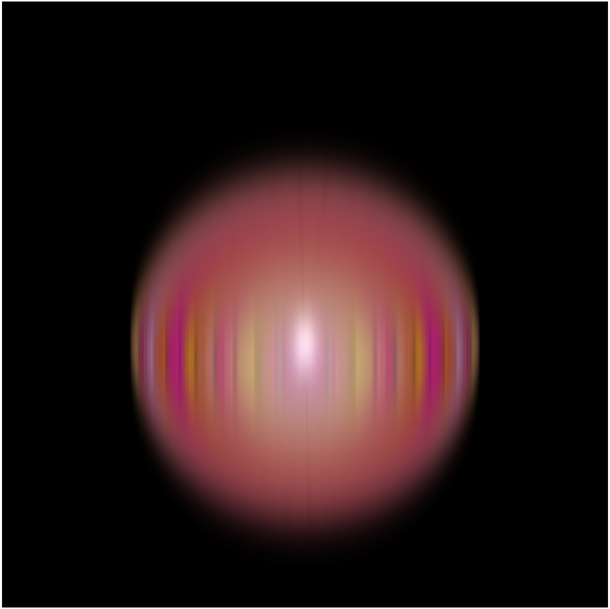
LightIntensity : Linear



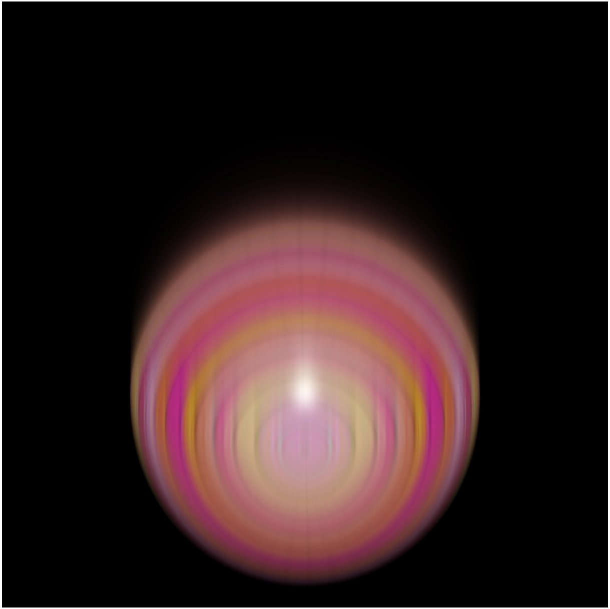
LightIntensity : Hat



LightIntensity : Gaussian

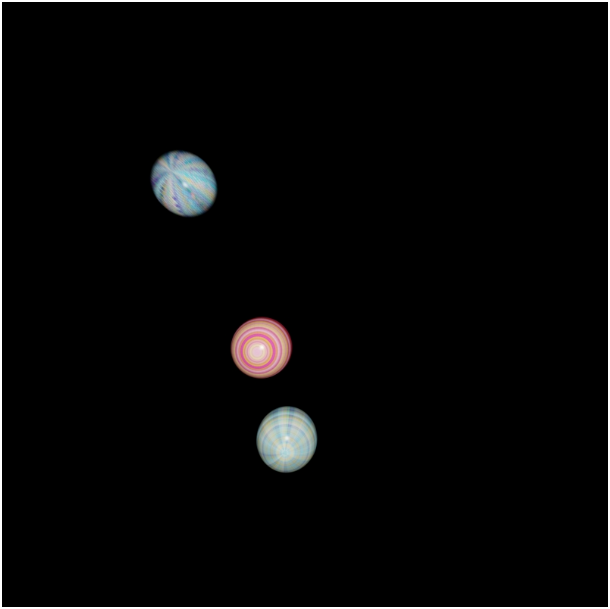


LightIntensity : Exponential

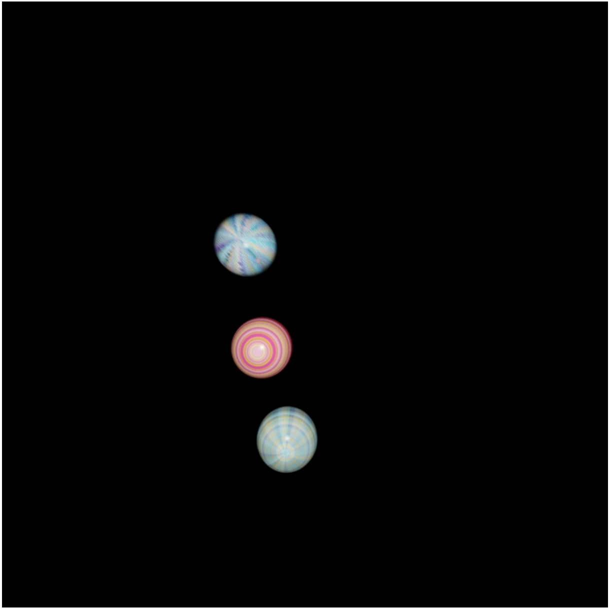


Part III: Additional Scenes

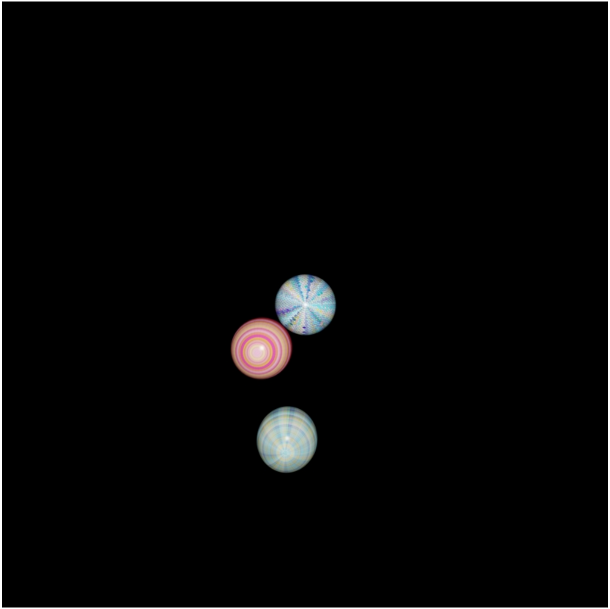
Multi Dynamic Objects



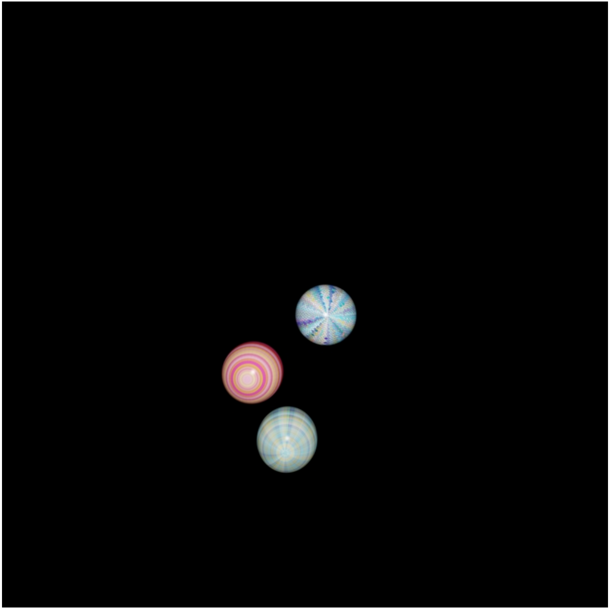
Multi Dynamic Objects



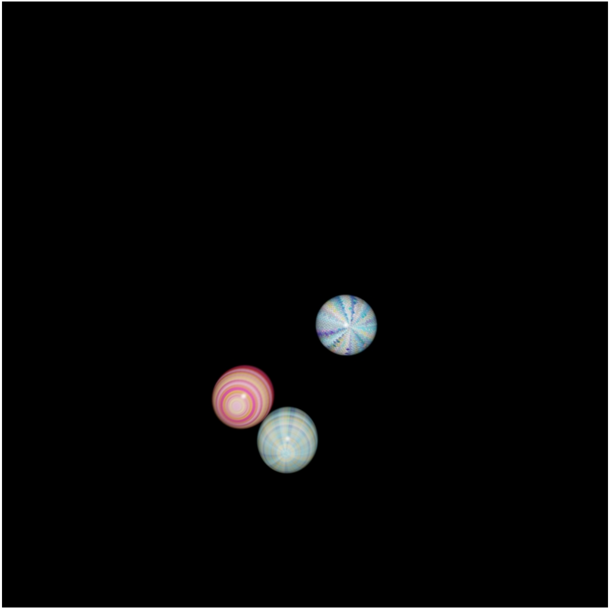
Multi Dynamic Objects



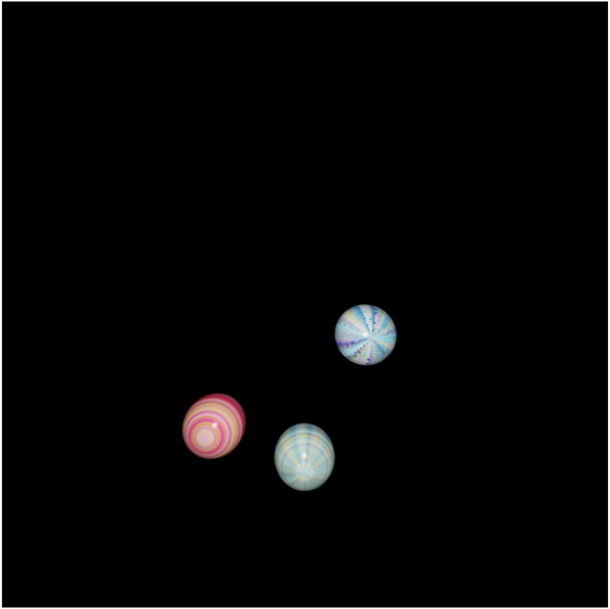
Multi Dynamic Objects



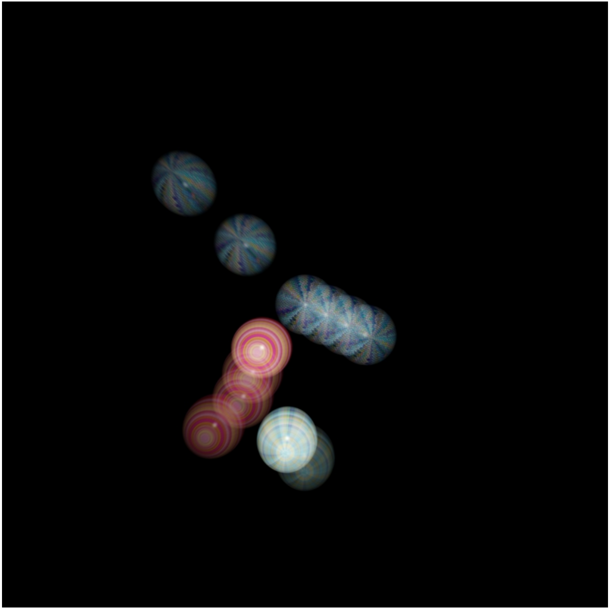
Multi Dynamic Objects



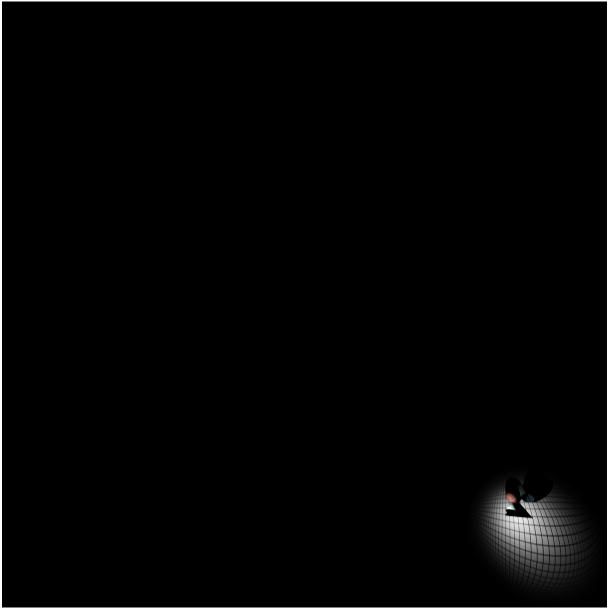
Multi Dynamic Objects



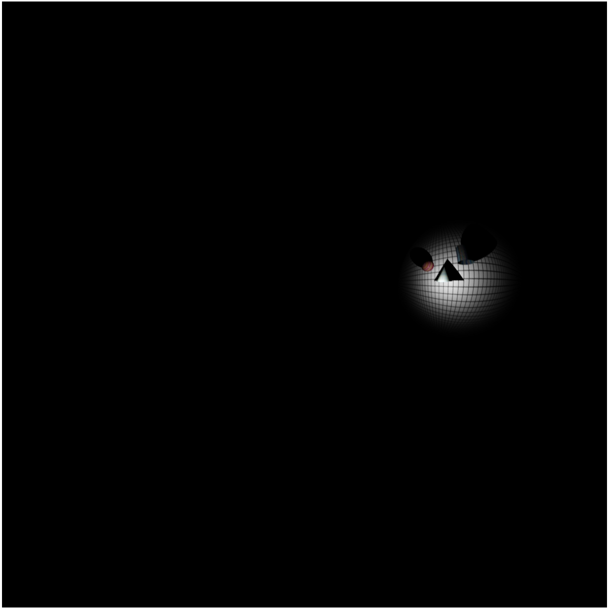
Multi Dynamic Objects



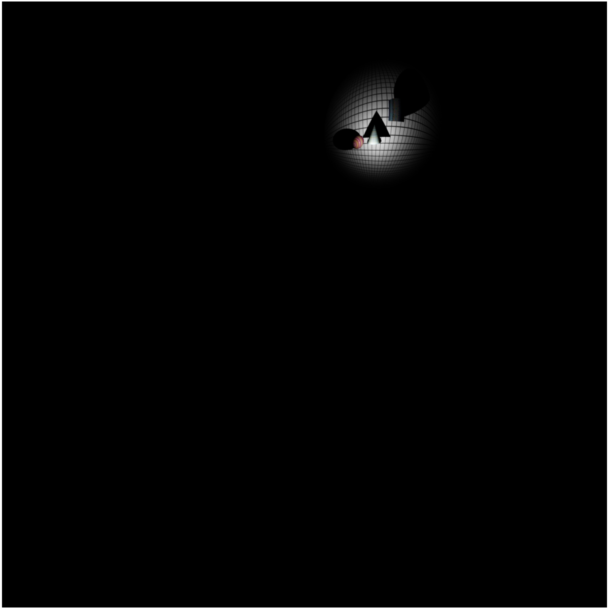
Specular Dynamic Object



Specular Dynamic Object

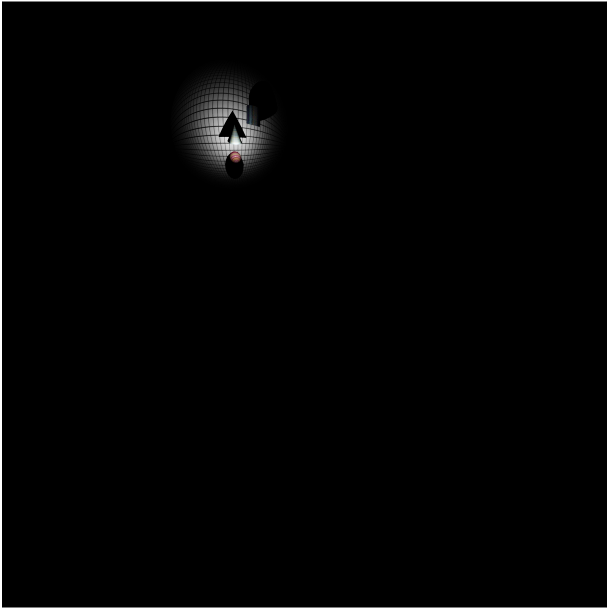


Specular Dynamic Object

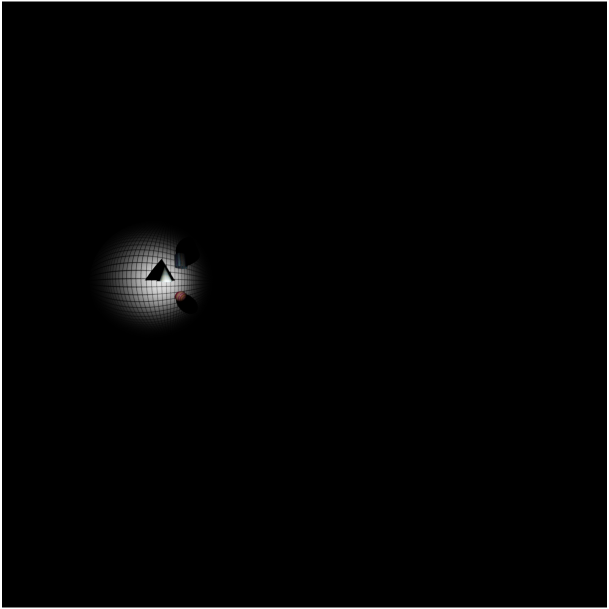


Specular Dynamic Object

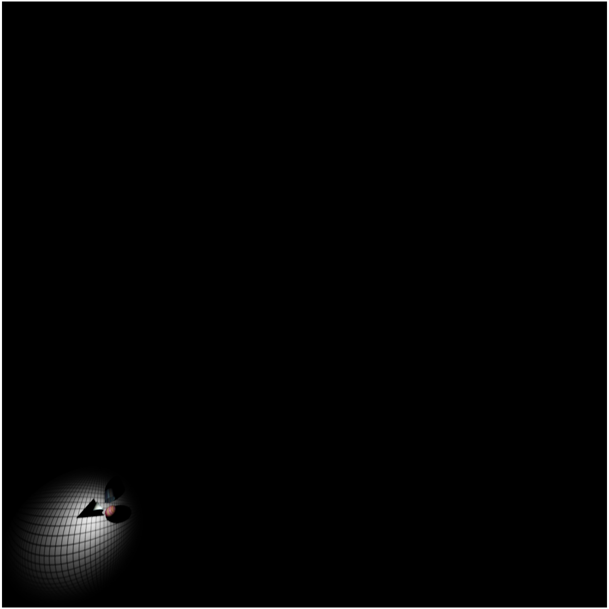
Specular Dynamic Object



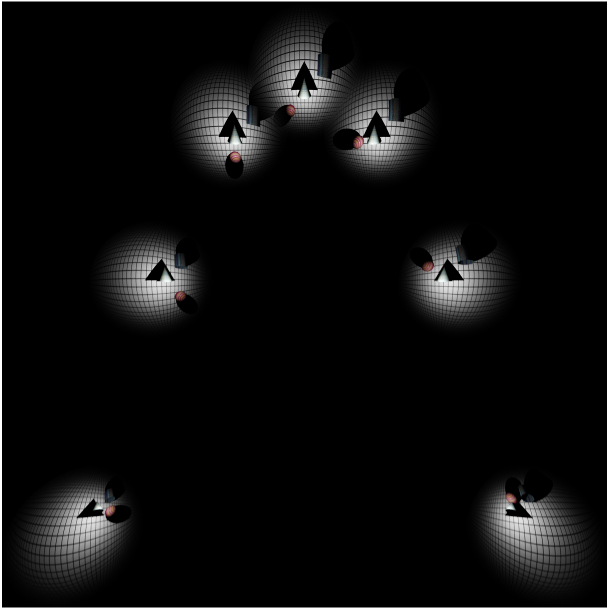
Specular Dynamic Object



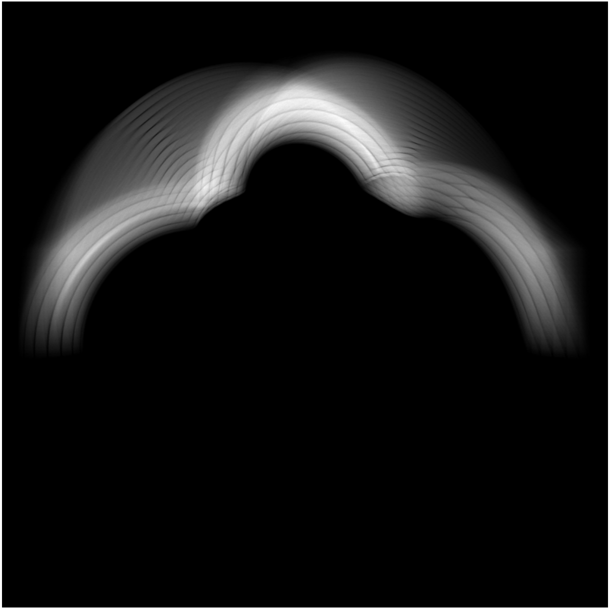
Specular Dynamic Object



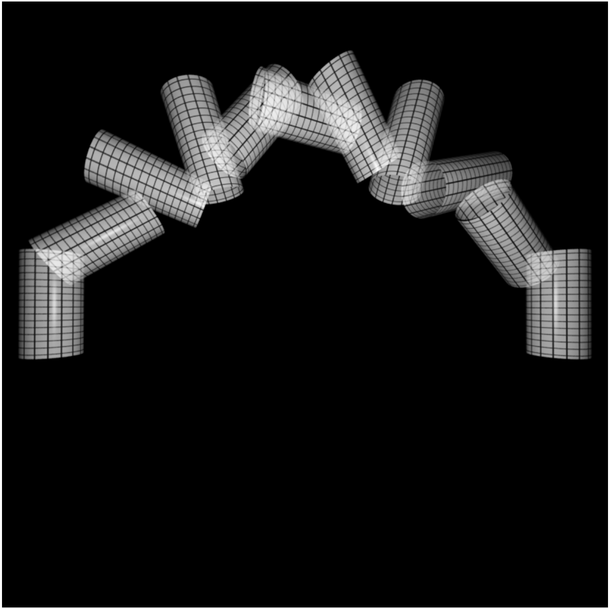
Specular Dynamic Object



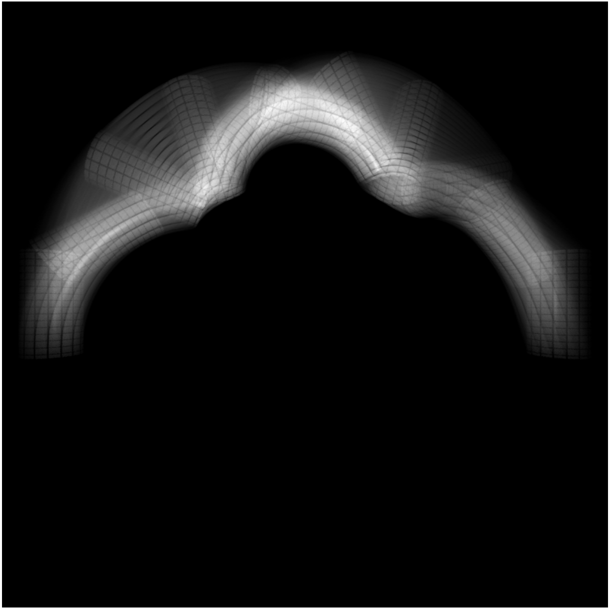
Light Configurations: Permanent



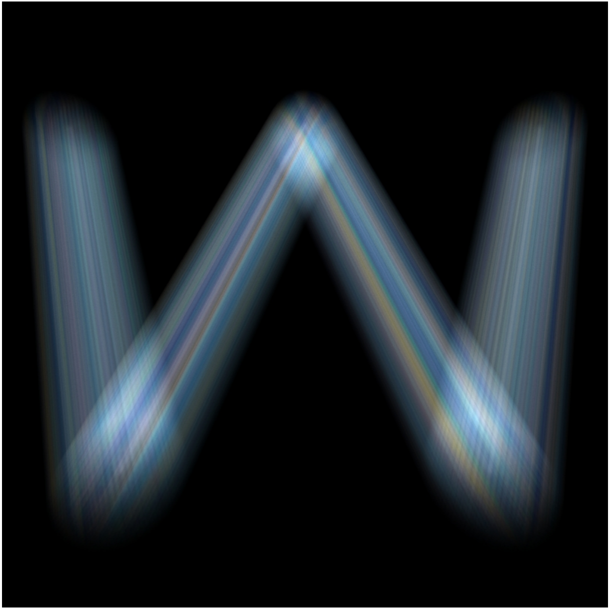
Light Configurations: Flash



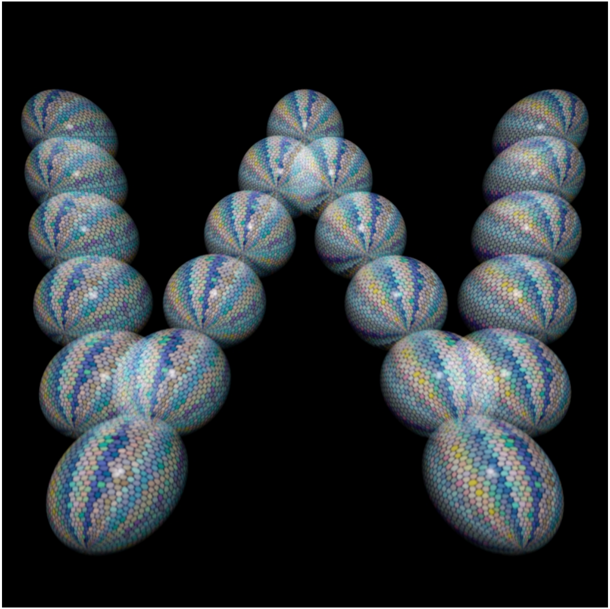
Light Configurations: Flash + Permanent



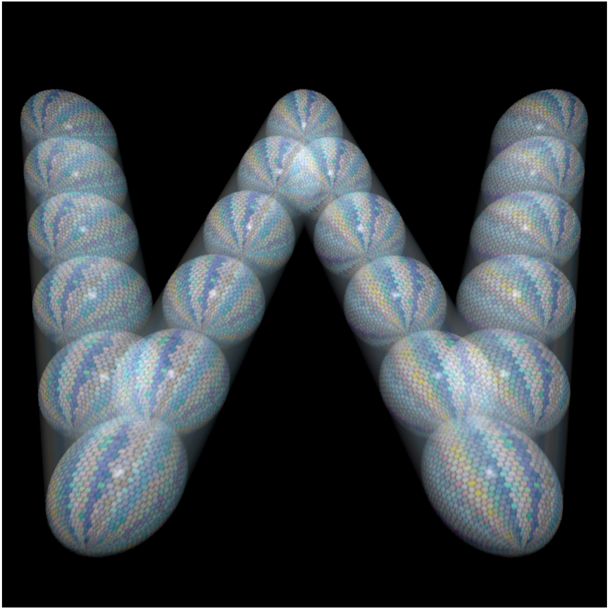
Light Configurations: Permanent



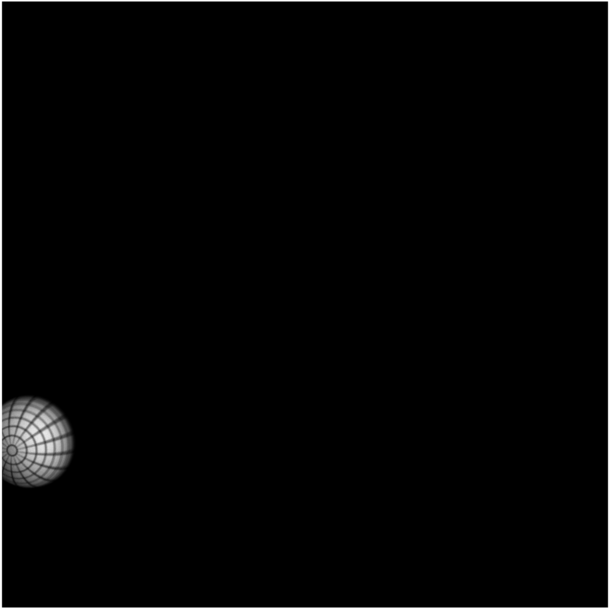
Light Configurations: Flash



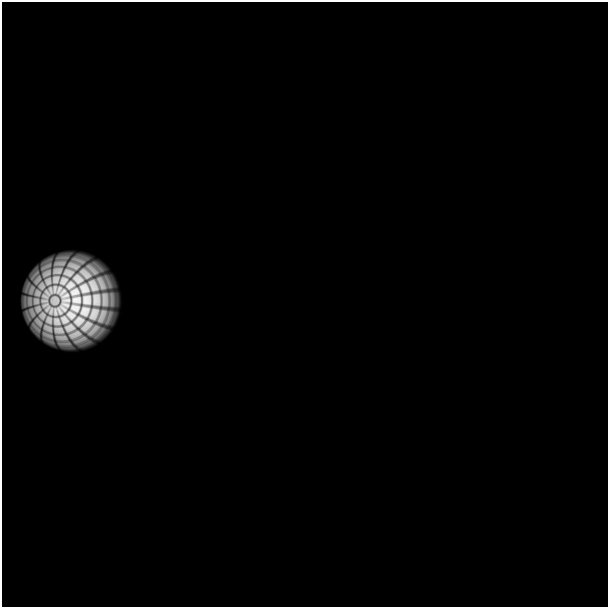
Light Configurations: Flash + Permanent



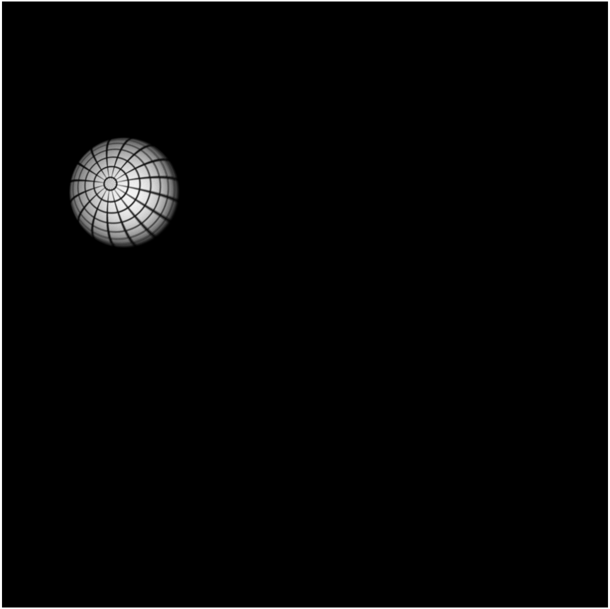
Realistic Camera and Motion



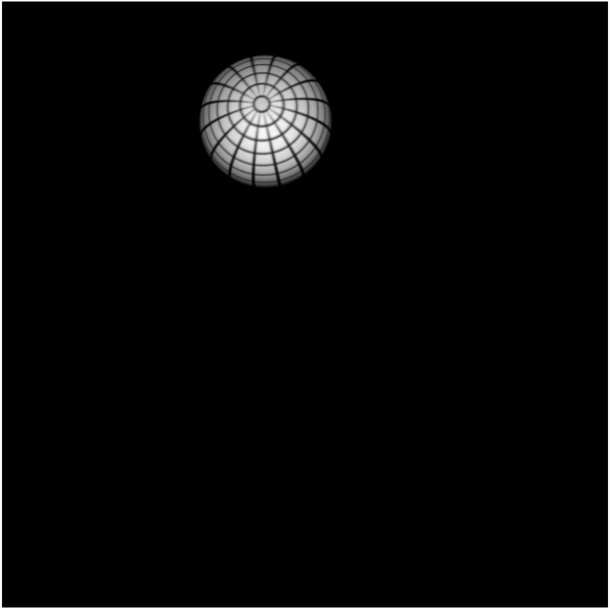
Realistic Camera and Motion



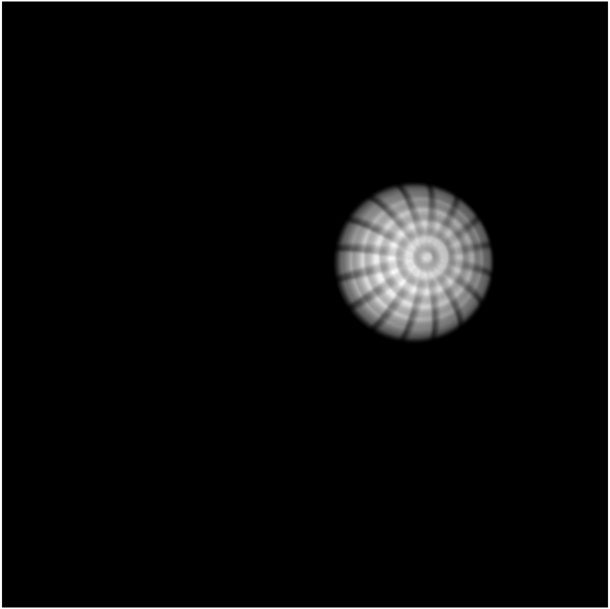
Realistic Camera and Motion



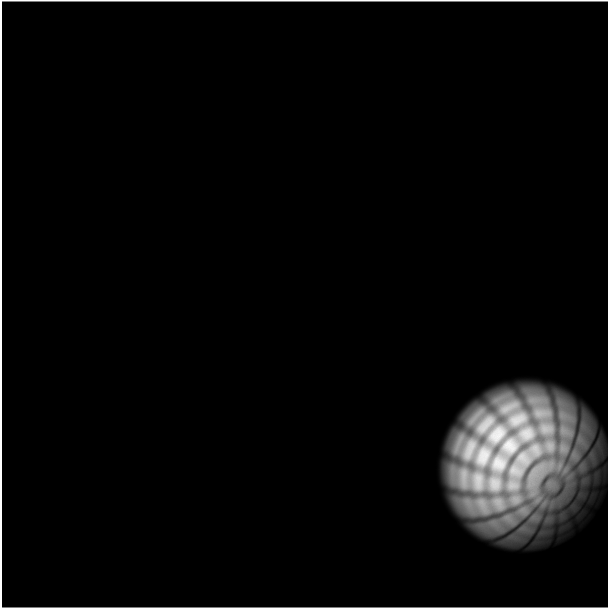
Realistic Camera and Motion



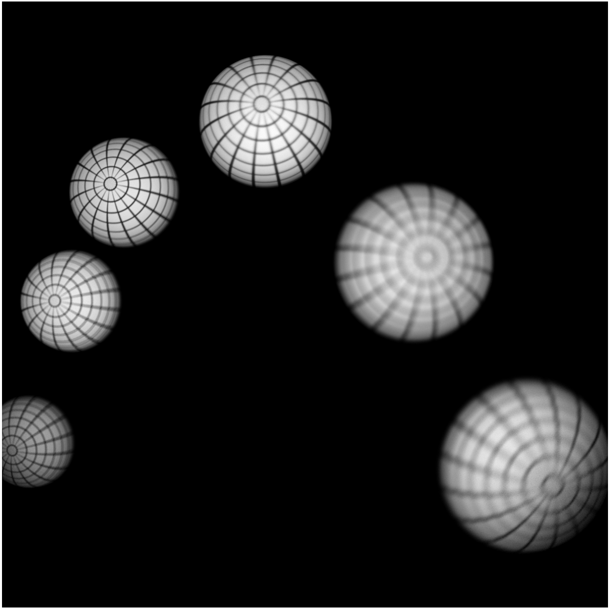
Realistic Camera and Motion



Realistic Camera and Motion



Realistic Camera and Motion



Conclusions

Conclusions

- Ray Tracing allow reconstruction of realistic dynamic scenes hardly attainable from other rendering techniques.

Conclusions

- Ray Tracing allow reconstruction of realistic dynamic scenes hardly attainable from other rendering techniques.
- Specific acceleration structures for dealing with dynamical objects improves the efficiency of the ray tracing process

Conclusions

- Ray Tracing allow reconstruction of realistic dynamic scenes hardly attainable from other rendering techniques.
- Specific acceleration structures for dealing with dynamical objects improves the efficiency of the ray tracing process
- Better sampling and reconstruction strategies are still required to reduce the computational cost of renderization.

Obrigado!

Cook, Robert L. and Porter, Thomas and Carpenter, Loren. *Distributed ray tracing*. SIGGRAPH 1984.

Hachisuka, Toshiya and Jarosz, Wojciech and Weistroffer, Richard Peter and Dale, Kevin and Humphreys, Greg and Zwicker, Matthias and Jensen, Henrik Wann. *Multidimensional adaptive sampling and reconstruction for ray tracing*. SIGGRAPH 2008.