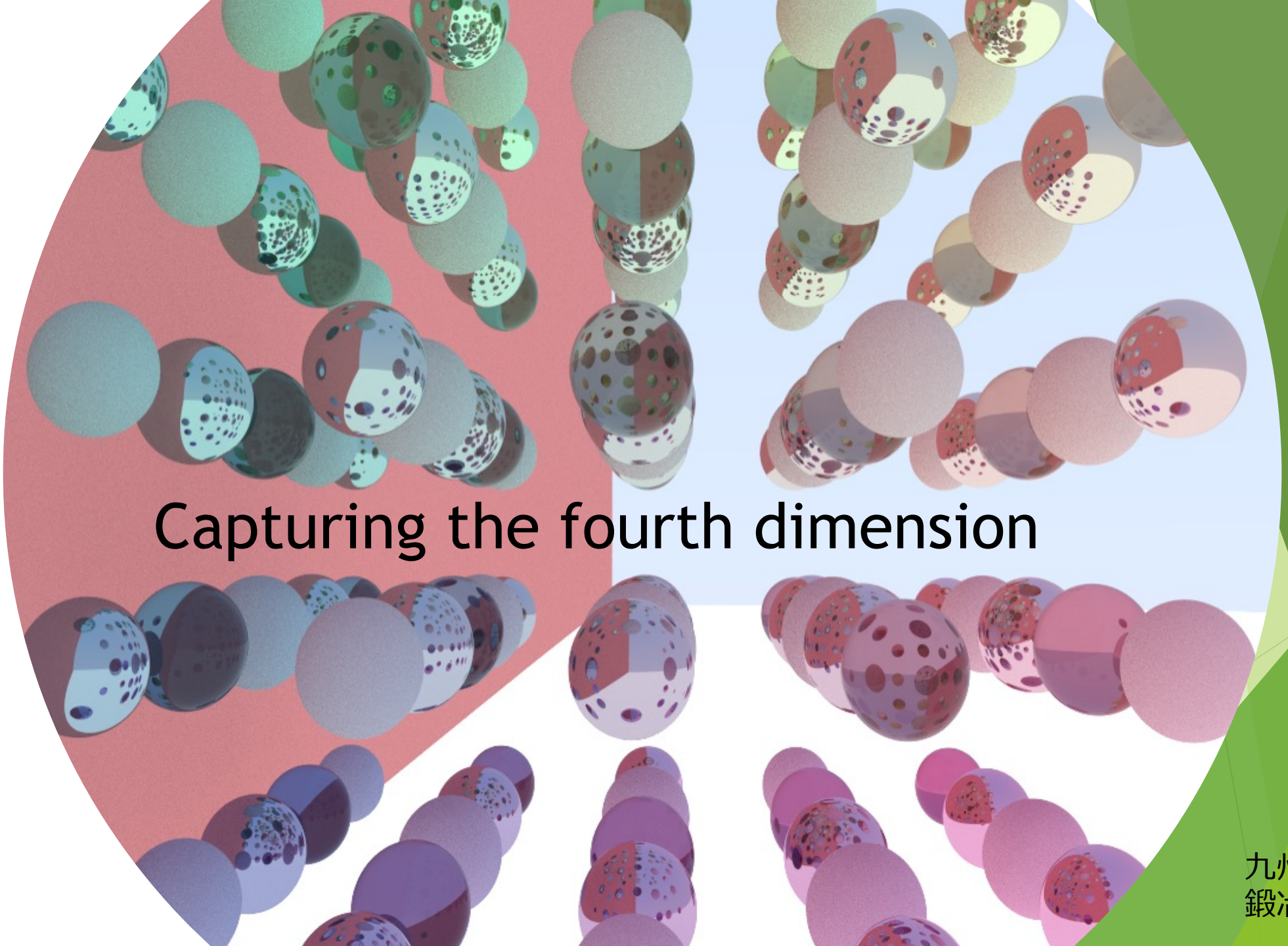


Capturing the fourth dimension

九州大学
鍛冶 静雄

- 前回のワークショップ中に開設されたポータルサイト

https://matheduvr.github.io

- Slack (招待制) 質問などこちらに書いて頂いても

matheduvr.slack.com
#2022imi

高次元知覚理解への問題設定 (Problem setting)

- **Objective & Task**:
  何を見たいのか．どんな対象の，いかなる性質？
  What (objects and their properties) do we want to “see”?
- **Method**:
  どうやって見るのか？
- **Evaluation**: 何をもって「高次元が見えた」と言えるのか？

A unwell-defined list of tasks

- Observe
  ４次元空間の中に置かれた図形(曲面など)を見る (embedded shape)
- Navigate & Orient
  ４次元空間の中を動き回る (ambient space)
- Understand
  位置を”知る”
  数える・測る
  より一般に位相や幾何に関する性質を理解
- Interact
  動かしたり組合わせたり変形したり

Ideas to encode an additional dimension

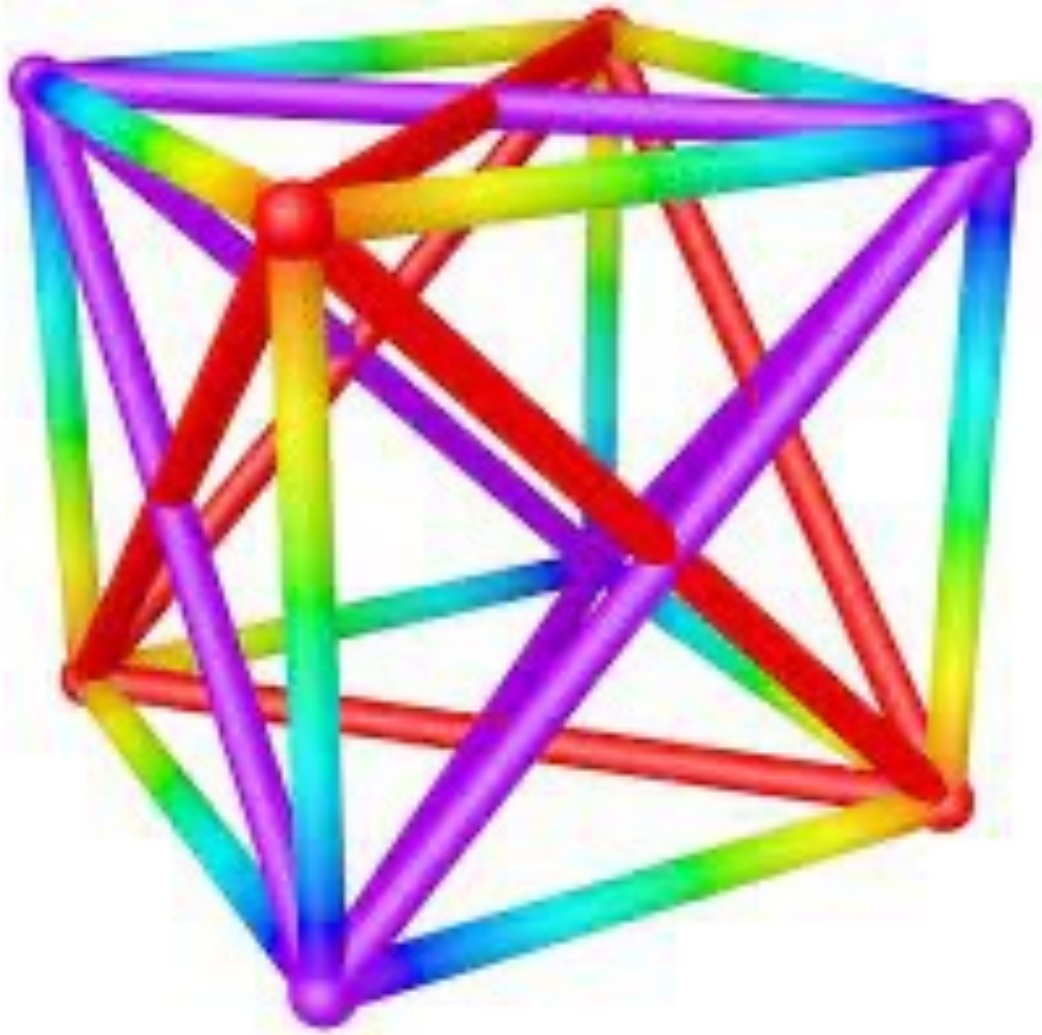
- Splitting colour and shape (”4D Draw” by Jeffrey Weeks <=
  The author of “The Shape of Space” ”Curved Space”)
- Splitting two eyes
  - binocular disparity 両眼視差 (“PolyVision”)
  - Binocular rivalry 視野闘争
- Splitting time window
  - Motion parallax 運動視差 (“PolyVision”)
  - Animation of slices, Morse theory
    (“4D Toys”, “Miegakure” by Marc ten Bosch)
- Splitting spatial window
  - Multiple projections (“PolyVision”)
- Interaction, Physics, Geometry
  - 4D kinematics (“4D Toys” by Marc ten Bosch)
  - 4D optics (“smallpt4d”)
  - 4D perspective drawing?

Principle 1
人間の持っている認識のためのリソースを何らかの形で分割して割り当てる
(Split and allocate human ability)

Principle 2
関係による手がかりを構成する
(Utilise relational cue)

どの方法がどの点で良いかを評価する
実験タスクや指標(ベンチマーク)の設計が重要

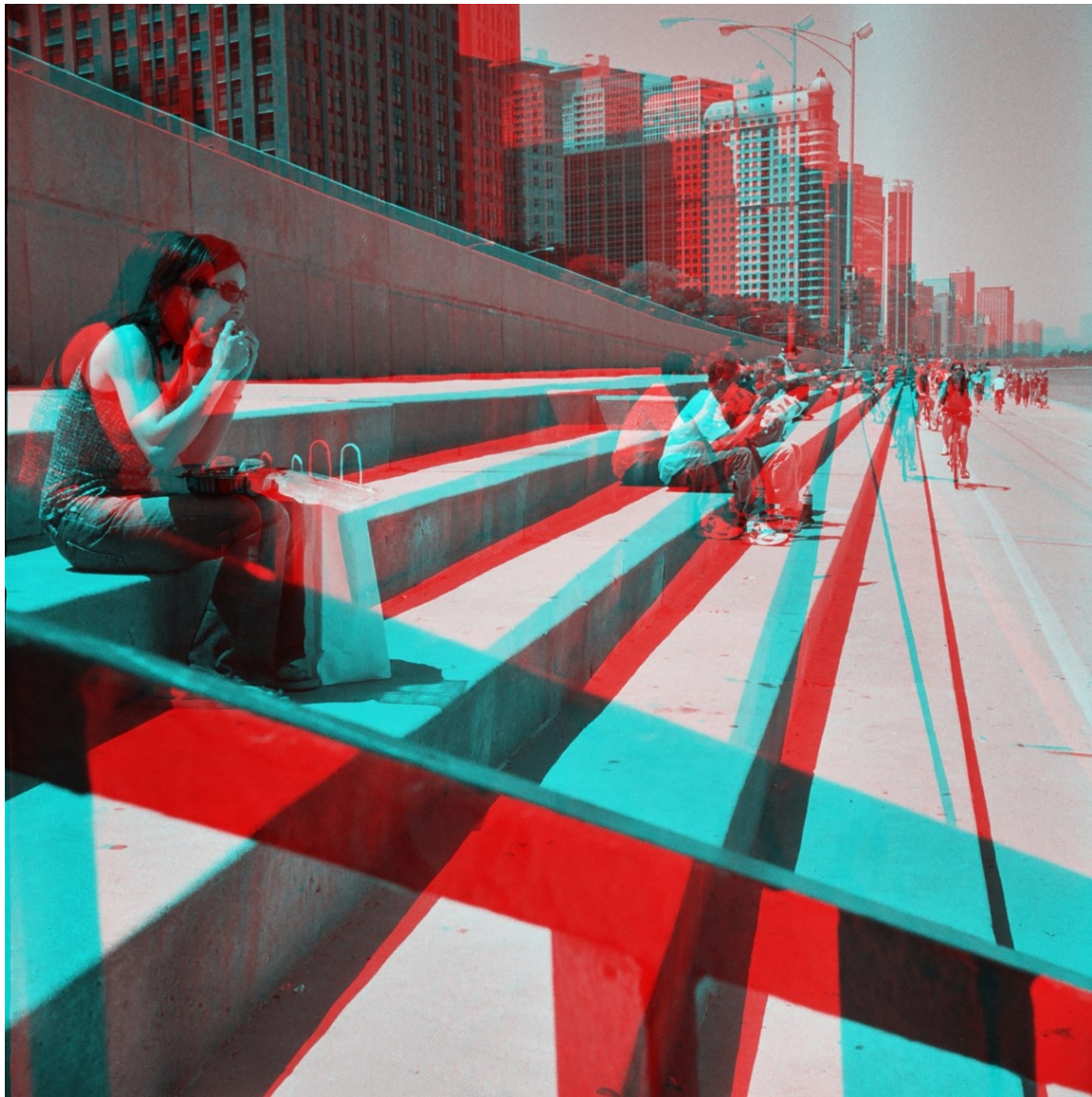
4D draw by J. Weeks



N. Ogawa, J. Shigeyama, T. Narumi, M. Hirose
Swinging 3D Lamps: A Projection Technique to Create 3D Illusions on a Static 2D Image
https://www.youtube.com/watch?v=Kh-xVL7q4cc

Binocular rivalry (赤青メガネで見ると，左右眼で違う情報が)

anaglyph

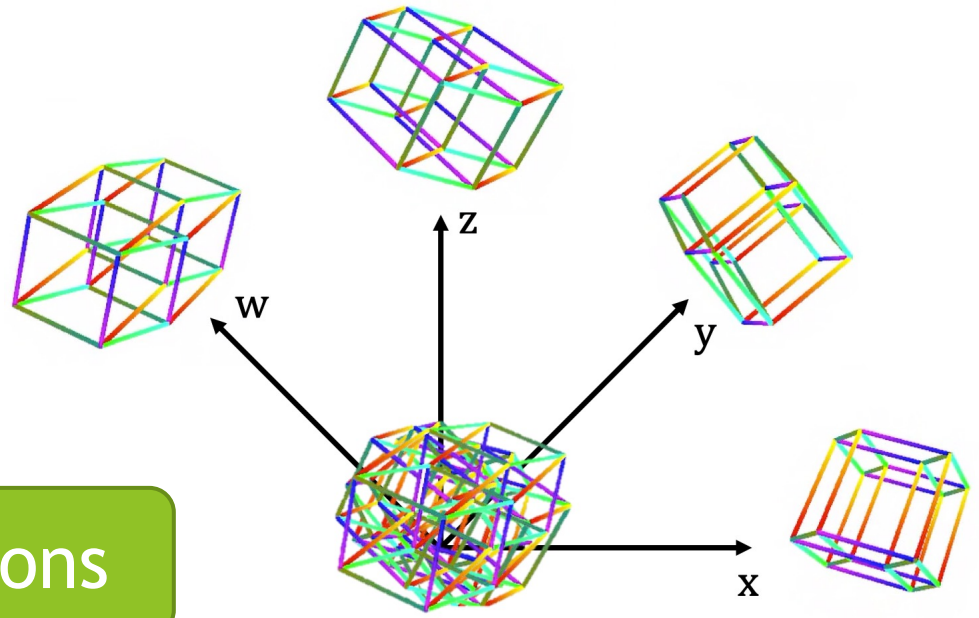


Images from Wikipedia

PolyVision

K. Matsumoto, N. Ogawa, H. Inou, S. Kaji, Y. Ishii, and M. Hirose, Polyvision: 4D Space Manipulation through Multiple Projections, SIGGRAPH Asia 2019 Emerging Technologies
https://www.youtube.com/watch?v=4PylcX8lnKo

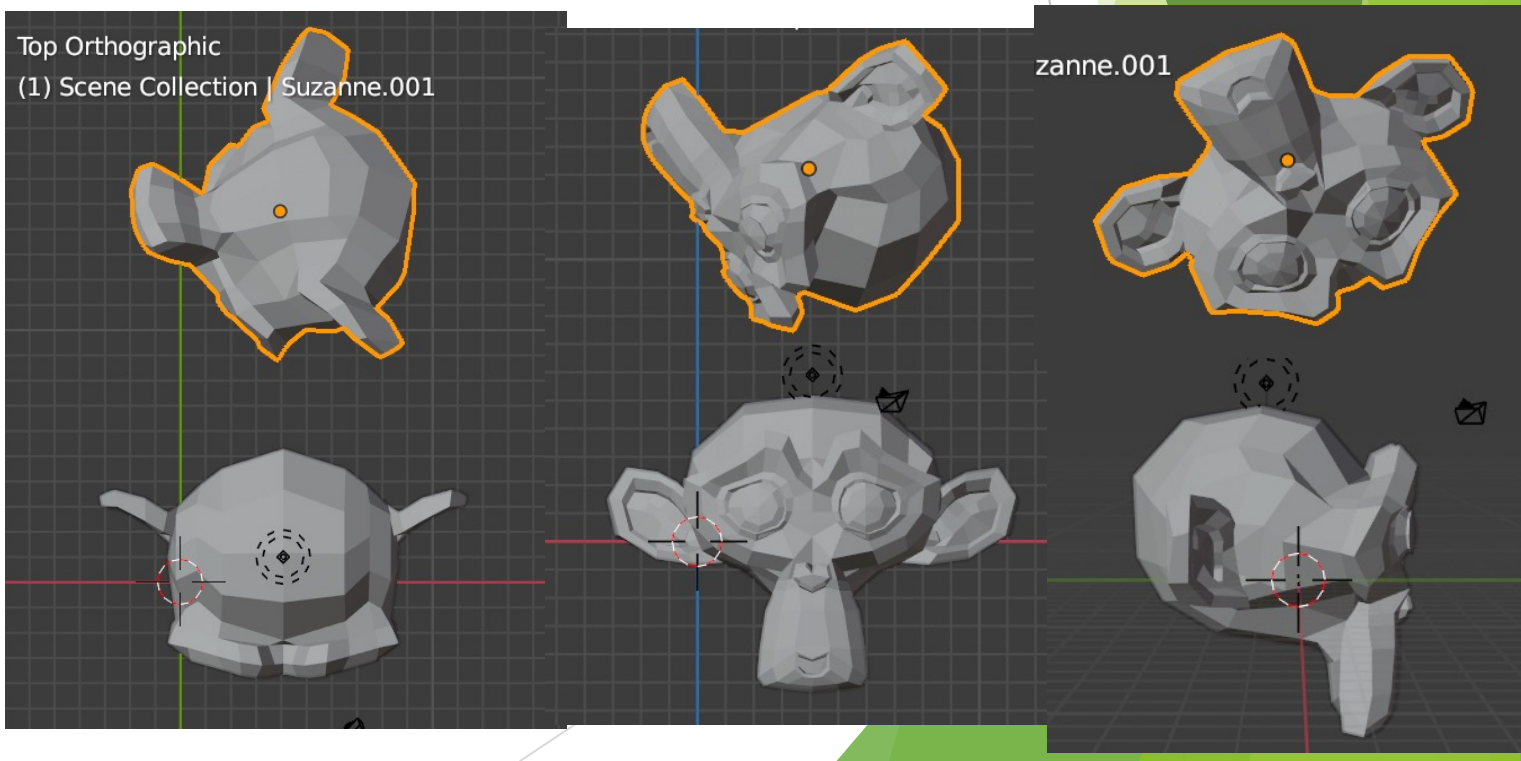
Key ideas of PolyVision



Concept

- 群盲象を評する
- 二人羽織
- 3D CG modeller

Multi-projections



Integrating
Multi-projection
with
Multi-consciousness

Possible Evaluation Benchmarks

- 同相(homeomorphic)か・isotope か判定する
- 高次の homotopy や homology の generators を認識する
- Mirror or not を判定
- Congruent or not を判定
- 面や穴の数を数える Count
- 合同な剛体を同じ向きに揃える Orient
- 積み木をお手本通りに並べる Place
- 空間中の任意の一点を指す Locate
- 迷路を解く Solve Maze
- 長さ・体積を比べる Measure & compare

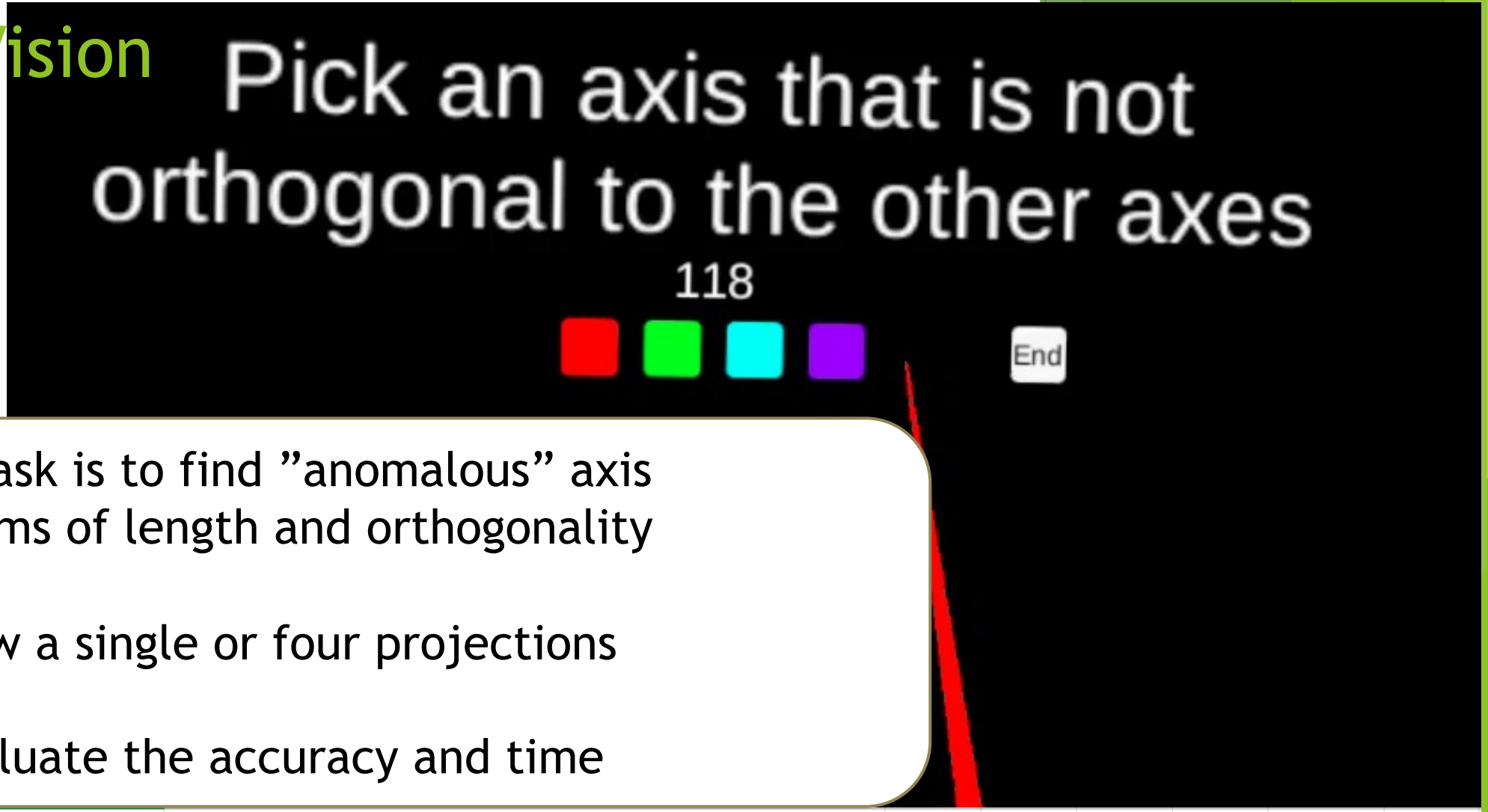
Passive 受動的
(操作を要求しない)

Active
能動的

評価指標(metric)
・正答率
・正答時間
・訓練による改善率

これらは３次元の心理実験の拡張で，幾何的情報把握を見ている．
４次元特有のタスクは作れるか？
Are there any 4D specific tasks?

Task in PolyVision



The task is to find ”anomalous” axis in terms of length and orthogonality

Show a single or four projections

Evaluate the accuracy and time

time	method	showOnly	length	matrix	indices	answer	result	endTime	confidence	difficulty
104.052	NonOrthogonalAxis	TRUE	(1.0, 1.0, 1.0, 1.0)	[(-0.5, -0.4, 0.4, 0.7),(0.0, 1.0, 0.0, 0.0),(0.0, 0.0, 1.0, 0.0),(0.0, 0.0, 0.0, 1.0)]	(3,1,0,2)	Y	FALSE	93.06425	1	1
207.67	NonOrthogonalAxis	FALSE	(1.0, 1.0, 1.0, 1.0)	[(-0.6, 0.5, 0.5, 0.3),(0.0, 1.0, 0.0, 0.0),(0.0, 0.0, 1.0, 0.0),(0.0, 0.0, 0.0, 1.0)]	(2,0,1,3)	Z	TRUE	101.697	1	1
268.376	NonOrthogonalAxis	TRUE	(1.0, 1.0, 1.0, 1.0)	[(-0.6, -0.5, 0.4, 0.4),(0.0, 1.0, 0.0, 0.0),(0.0, 0.0, 1.0, 0.0),(0.0, 0.0, 0.0, 1.0)]	(2,1,0,3)	Z	TRUE	58.75693	3	1
44.5413	OneDifferentLength	TRUE	(0.7, 1.0, 1.0, 1.0)	[(1.0, 0.0, 0.0, 0.0),(0.0, 1.0, 0.0, 0.0),(0.0, 0.0, 1.0, 0.0),(0.0, 0.0, 0.0, 1.0)]	(1,2,3,0)	Y	TRUE	28.11122	1	1
95.4732	OneDifferentLength	TRUE	(1.3, 1.0, 1.0, 1.0)	[(1.0, 0.0, 0.0, 0.0),(0.0, 1.0, 0.0, 0.0),(0.0, 0.0, 1.0, 0.0),(0.0, 0.0, 0.0, 1.0)]	(0,2,3,1)	X	TRUE	46.61589	1	1
106.751	OneDifferentLength	TRUE	(1.4, 1.0, 1.0, 1.0)	[(1.0, 0.0, 0.0, 0.0),(0.0, 1.0, 0.0, 0.0),(0.0, 0.0, 1.0, 0.0),(0.0, 0.0, 0.0, 1.0)]	(2,1,0,3)	Z	TRUE	6.822578	1	1
133.387	OneDifferentLength	TRUE	(0.7, 1.0, 1.0, 1.0)	[(1.0, 0.0, 0.0, 0.0),(0.0, 1.0, 0.0, 0.0),(0.0, 0.0, 1.0, 0.0),(0.0, 0.0, 0.0, 1.0)]	(2,0,1,3)	X	FALSE	23.66998	1	1

Interaction in 4D: rigid body motion

- 四元数と４次元ユークリッド空間を同一視すると

$$\mathbb{R}^4 \simeq \mathbb{H}$$

四元数の掛け算(非可換)により、右掛け算と左掛け算で変換を定義できる

特に、絶対値１の四元数に制限して、次のパラメトリゼーションを得る

$$\mathbb{H}_1 = \{q \in \mathbb{H} \mid q\bar{q} = 1\}$$

$$\mathbb{H}_1 \times \mathbb{H}_1 \to SO(4)$$

$$(q_1, q_2) \mapsto (v \mapsto q_1 v \bar{q}_2)$$

これは二重被覆(2対1写像)となっている。
特に、自由度は６

平行移動も加えた合同変換(自由度10)はどうやって指定する？
(geometric algebra?)

Wait, why rigid body motion?

- 着目したい性質は、本当に回転不変か？
  What we want to see may not be rotation invariant.

$\mathbb{C}^2 \simeq \mathbb{R}^4$ では U(2) or SU(2) の方が自然？

Unitary transformation invariance
may be more relevant

さらに metric を変えると (局所的には)

$R^{3,1}$ローレンツ空間(=2次エルミート行列) $O(3,1) \simeq SL(2;\mathbb{C})$

$R^{2,2}$ (=2次実行列) $O(2,2) \simeq SL(2) \times SL(2)$

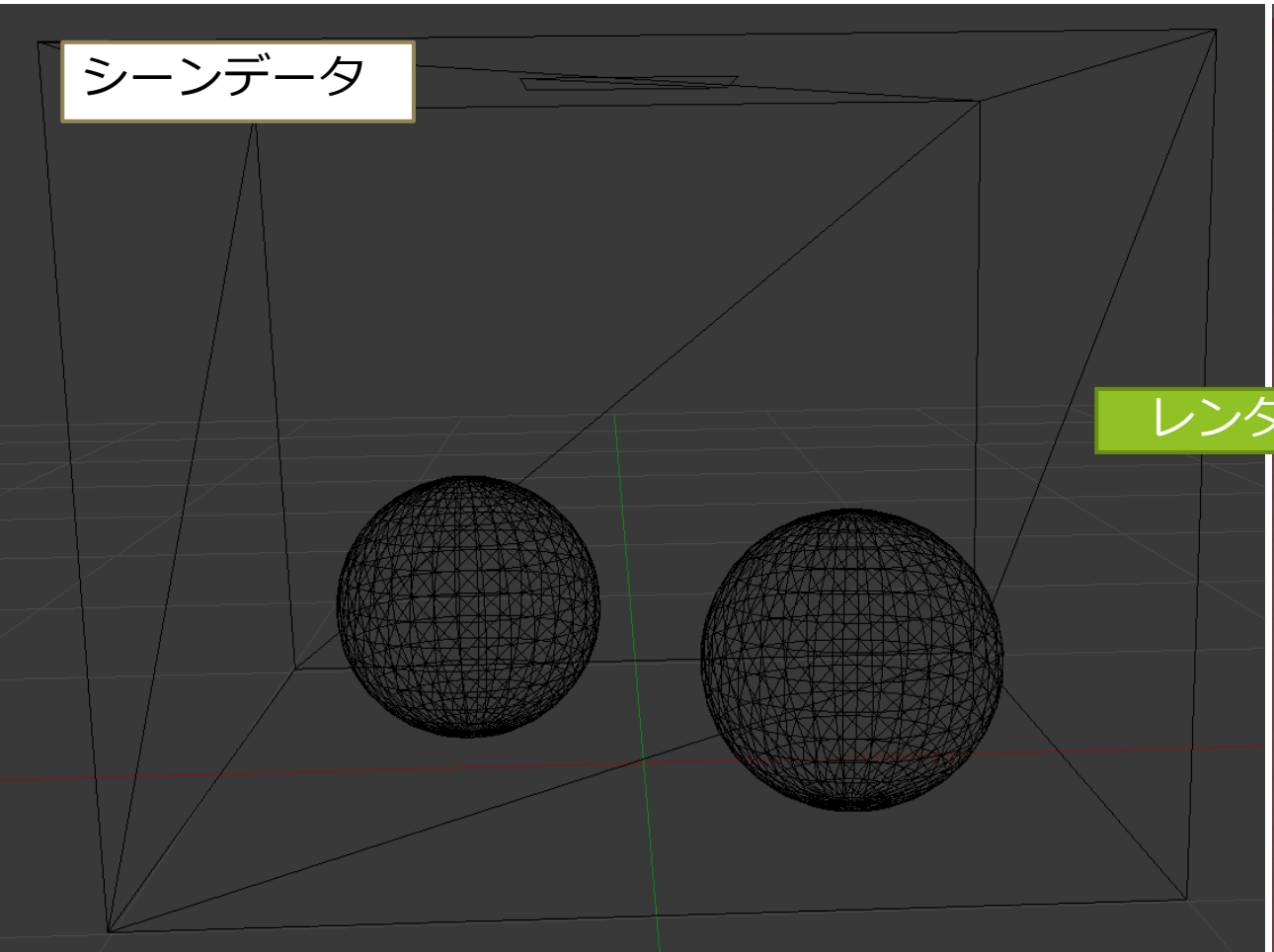
インタラクションにどの変換を紐づけるかは
どんな対象を見たいかに依存する！

Rendering – smallpt4d

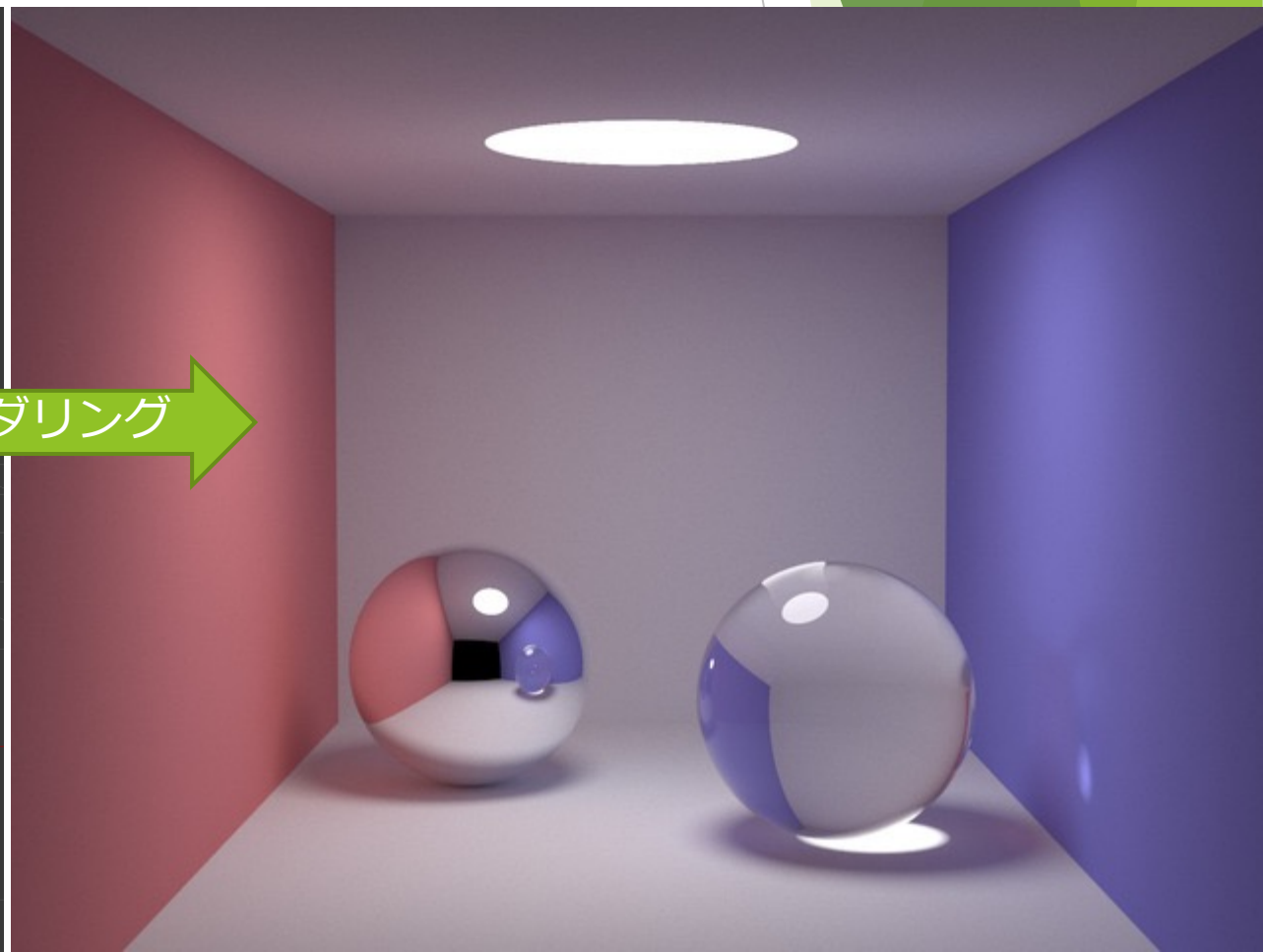
Source codes and videos available at
https://github.com/shizuo-kaji/smallpt4d

Rendering in 3D

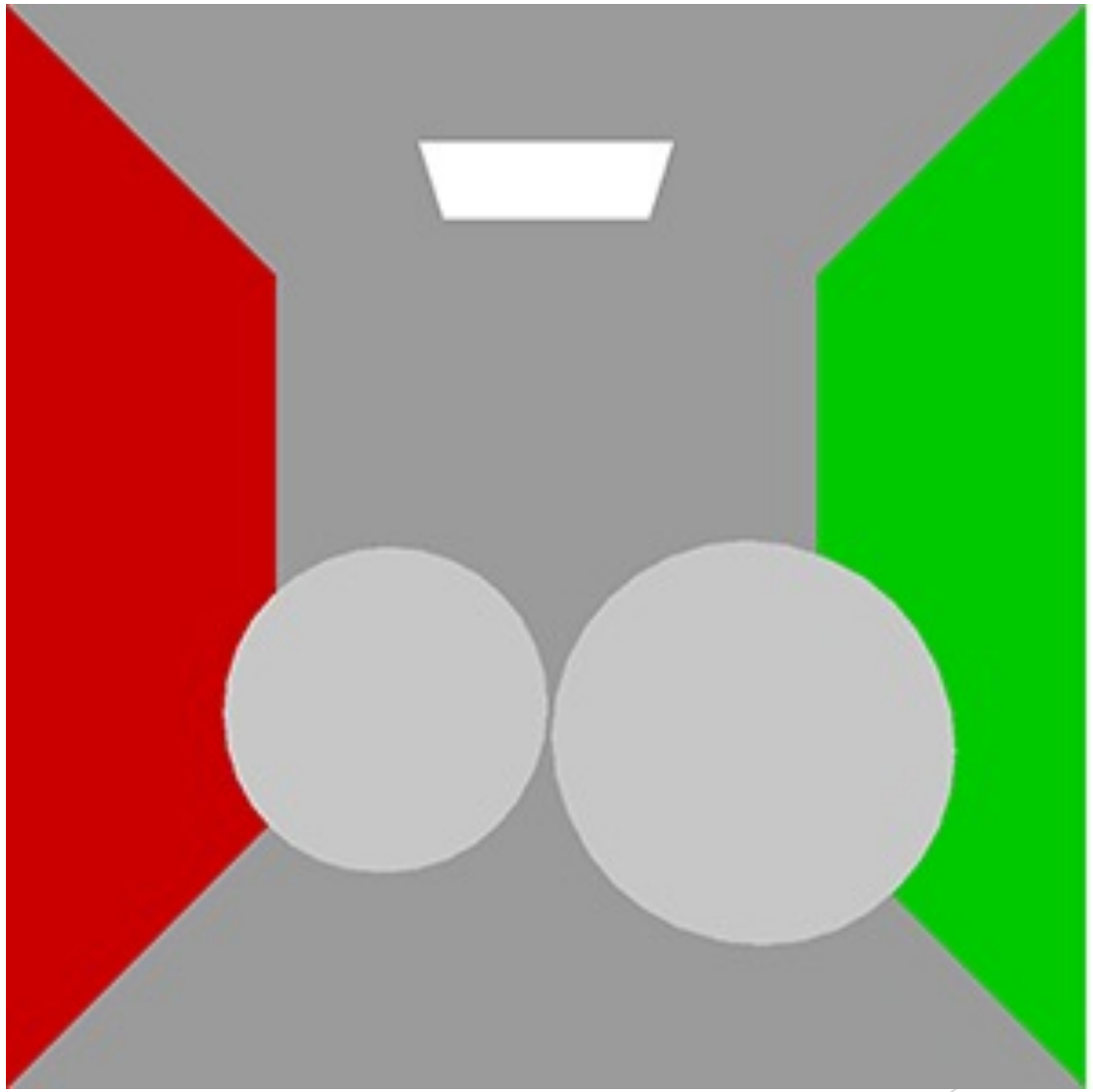
3D scene data needs to be *rendered* to produce a 2D image



レンダリング



If we ignore optics…



3次元シーンから2次元の絵を得る

Objects:
$\{A_i \subset R^3\}|$

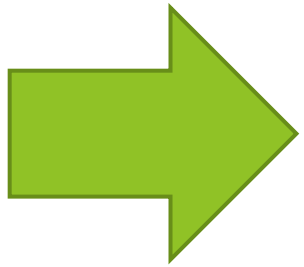
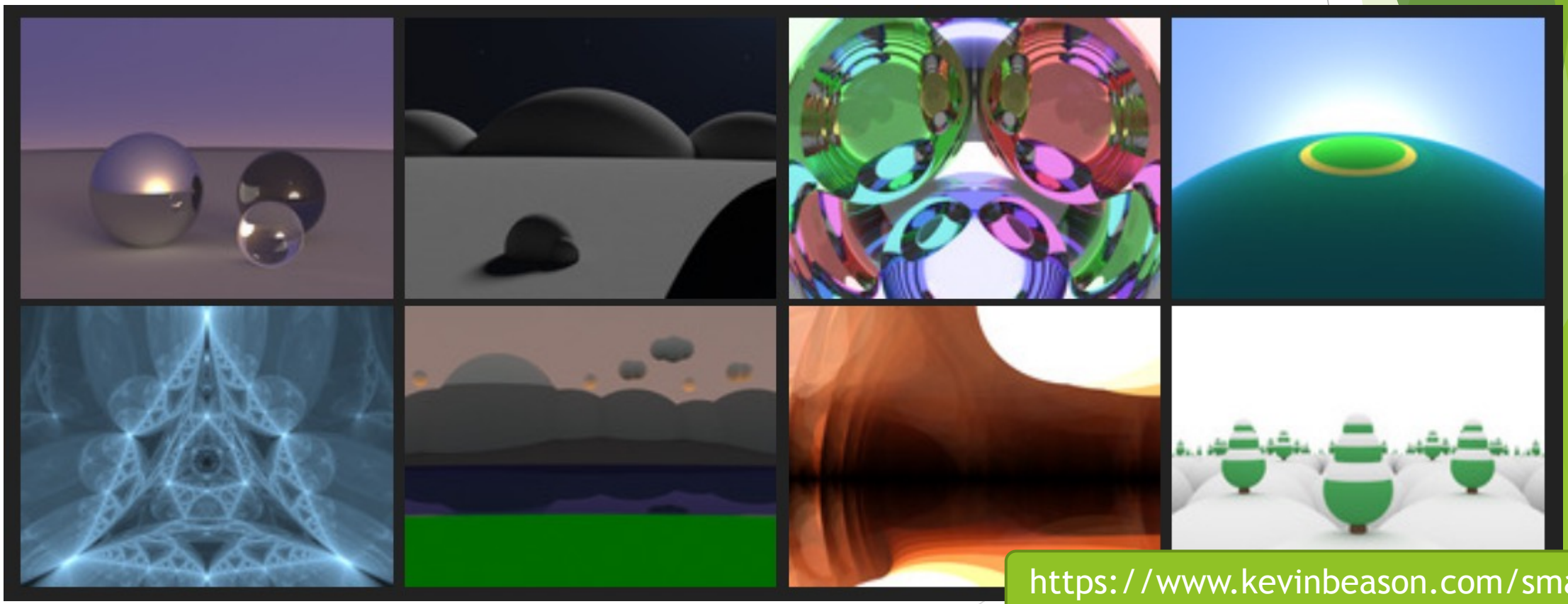


Image : $R^2 \rightarrow R^3$
RGB 3 channel

How to define this mapping?

smallpt

- written by Kevin Beason
- 99 line in C++
- Monte Carlo path tracing with Russian roulette path termination
- scenes must consist of spheres

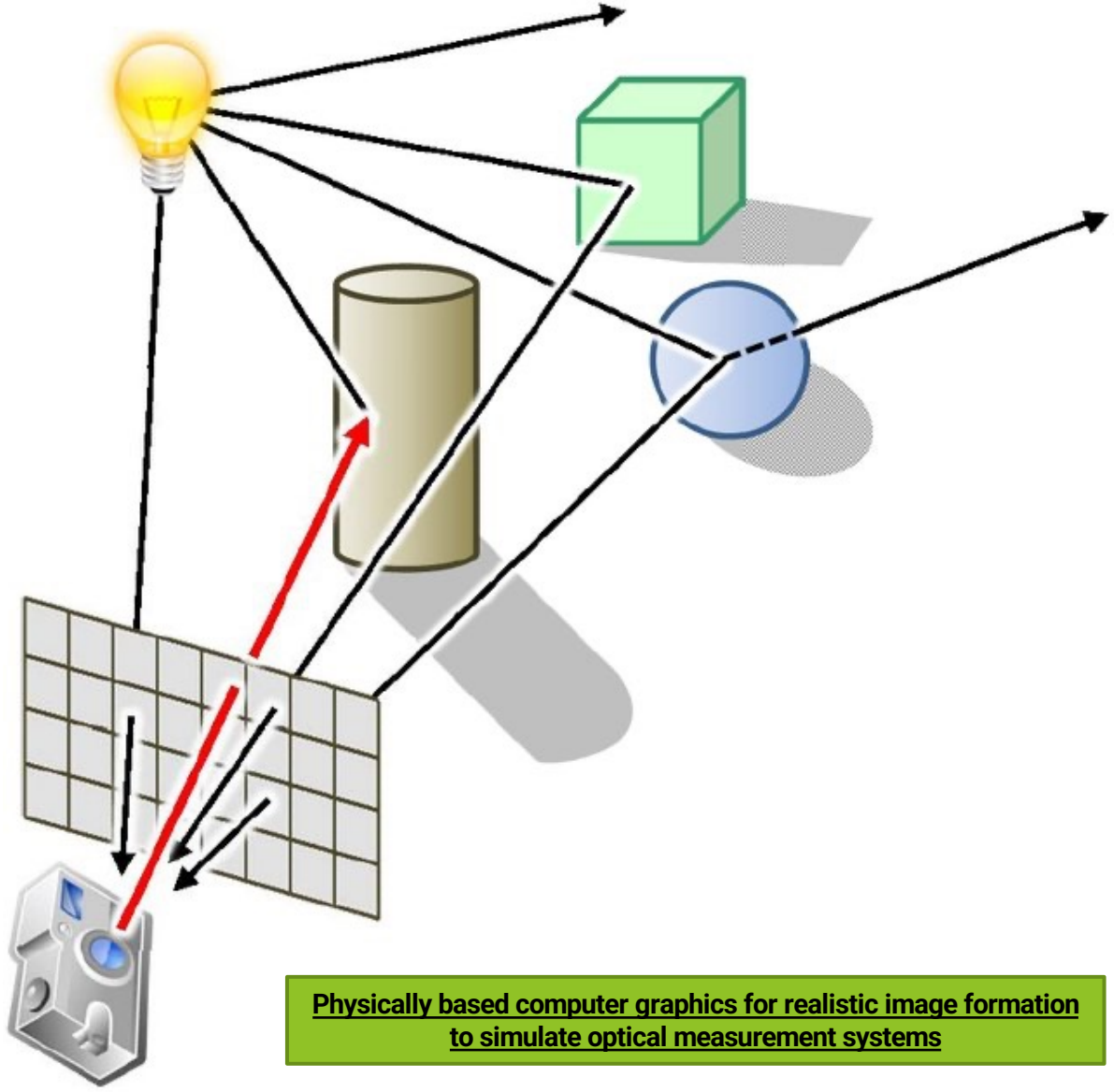


https://www.kevinbeason.com/smallpt/

smallpt4d: an extension for 4D scenes with binocular rendering

Optics for tracing light paths

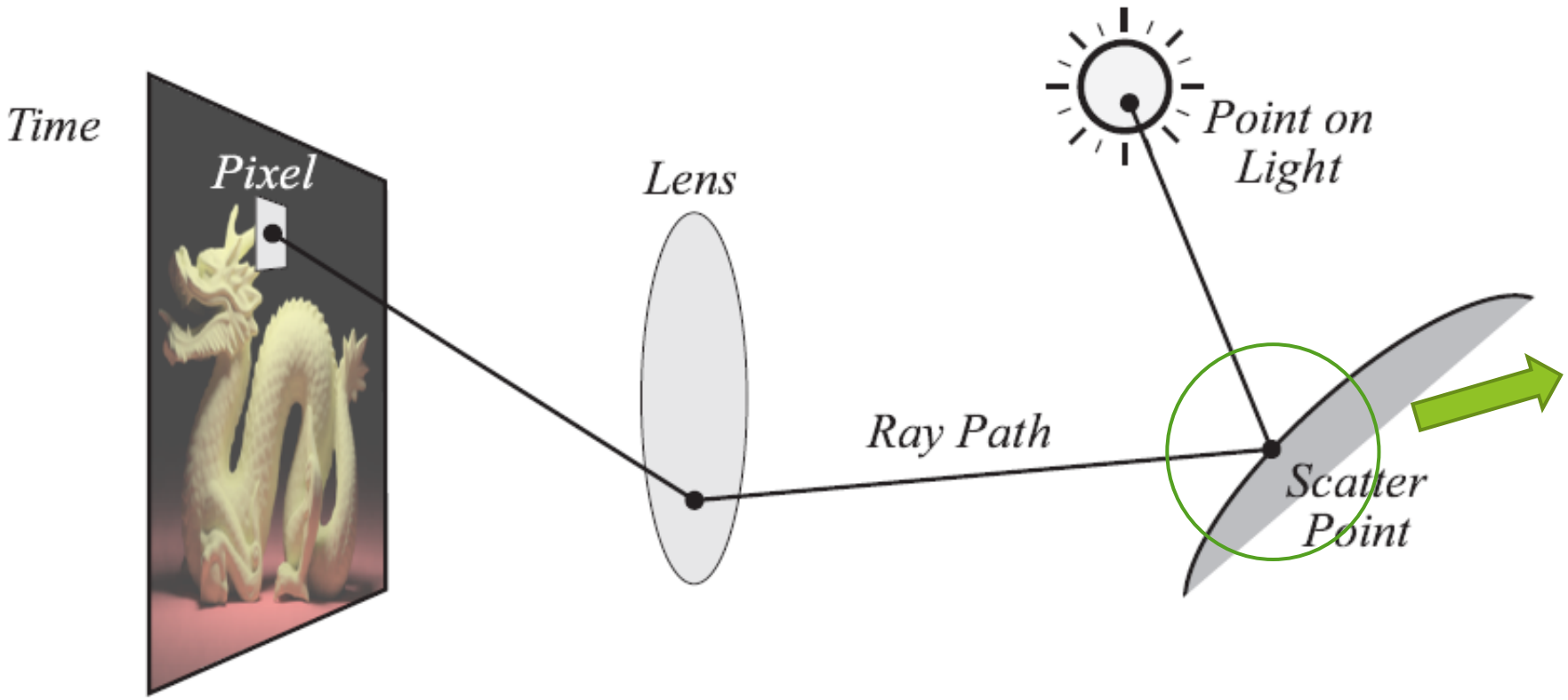
- Scene = Objects, Light, Camera
- Object = Geometry, Reflection/ refraction characteristics, Colour
- Light = Geometry, Colour, Intensity
- Camera = Origin, Screen

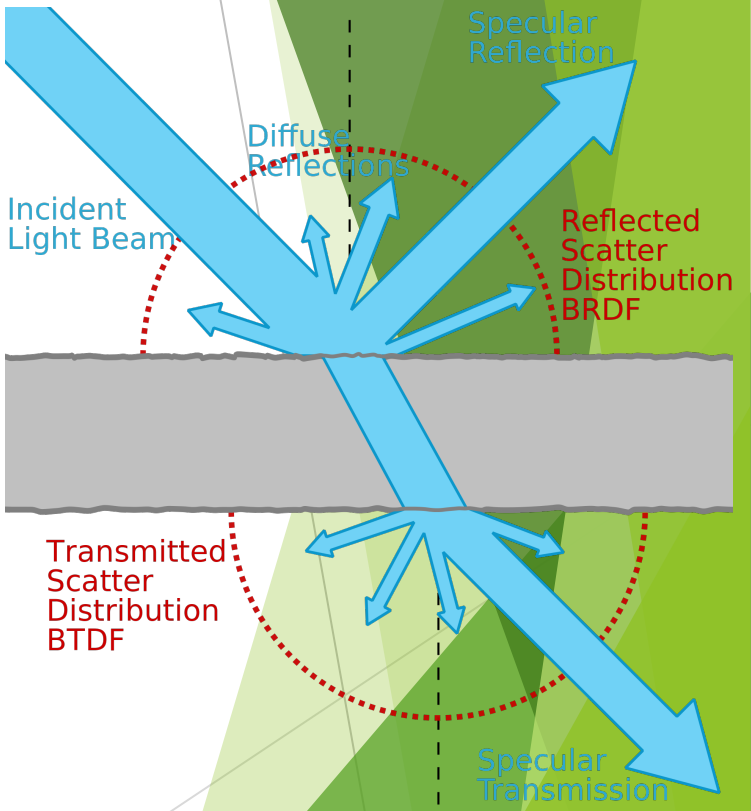


Physically based computer graphics for realistic image formation to simulate optical measurement systems

Light path

from the slides by Dr David Cline explaining smallpt
Available at https://www.kevinbeason.com/smallpt/





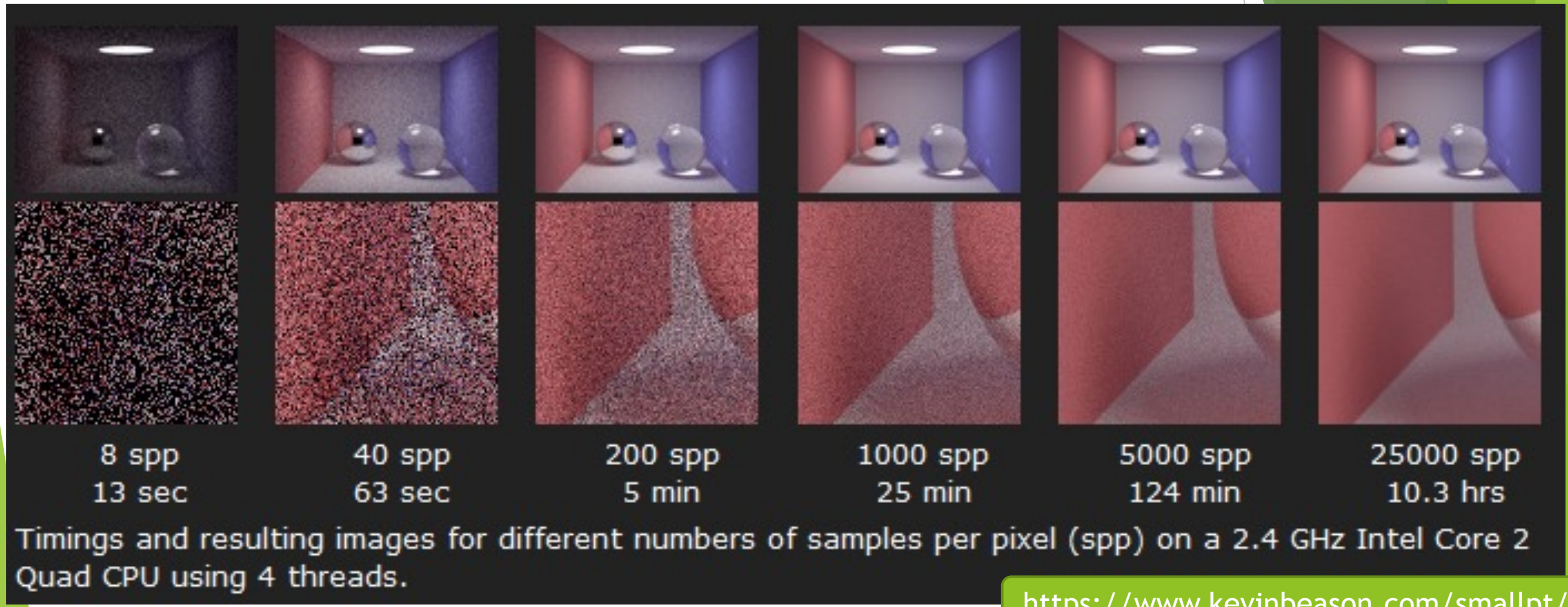
Path Tracing Algorithm

from the slides by Dr David Cline explaining smallpt

Algorithm 3 Path Tracing Main Loop

```
1:  for each pixel (i,j) do
2:      Vec3 C = 0
3:      for (k=0; k < samplesPerPixel; k++) do
4:          Create random ray in pixel:
5:              Choose random point on lens P_lens
6:              Choose random point on image plane P_image
7:              D = normalize(P_image − P_lens)
8:              Ray ray = Ray(P_lens, D)
9:          castRay(ray, isect)
10:         if the ray hits something then
11:             C += radiance(ray, isect, 0)
12:         else
13:             C += backgroundColor(D)
14:         end if
15:     end for
16:     image(i,j) = C / samplesPerPixel
17: end for
```

Convergence



Timings and resulting images for different numbers of samples per pixel (spp) on a 2.4 GHz Intel Core 2 Quad CPU using 4 threads.

https://www.kevinbeason.com/smallpt/

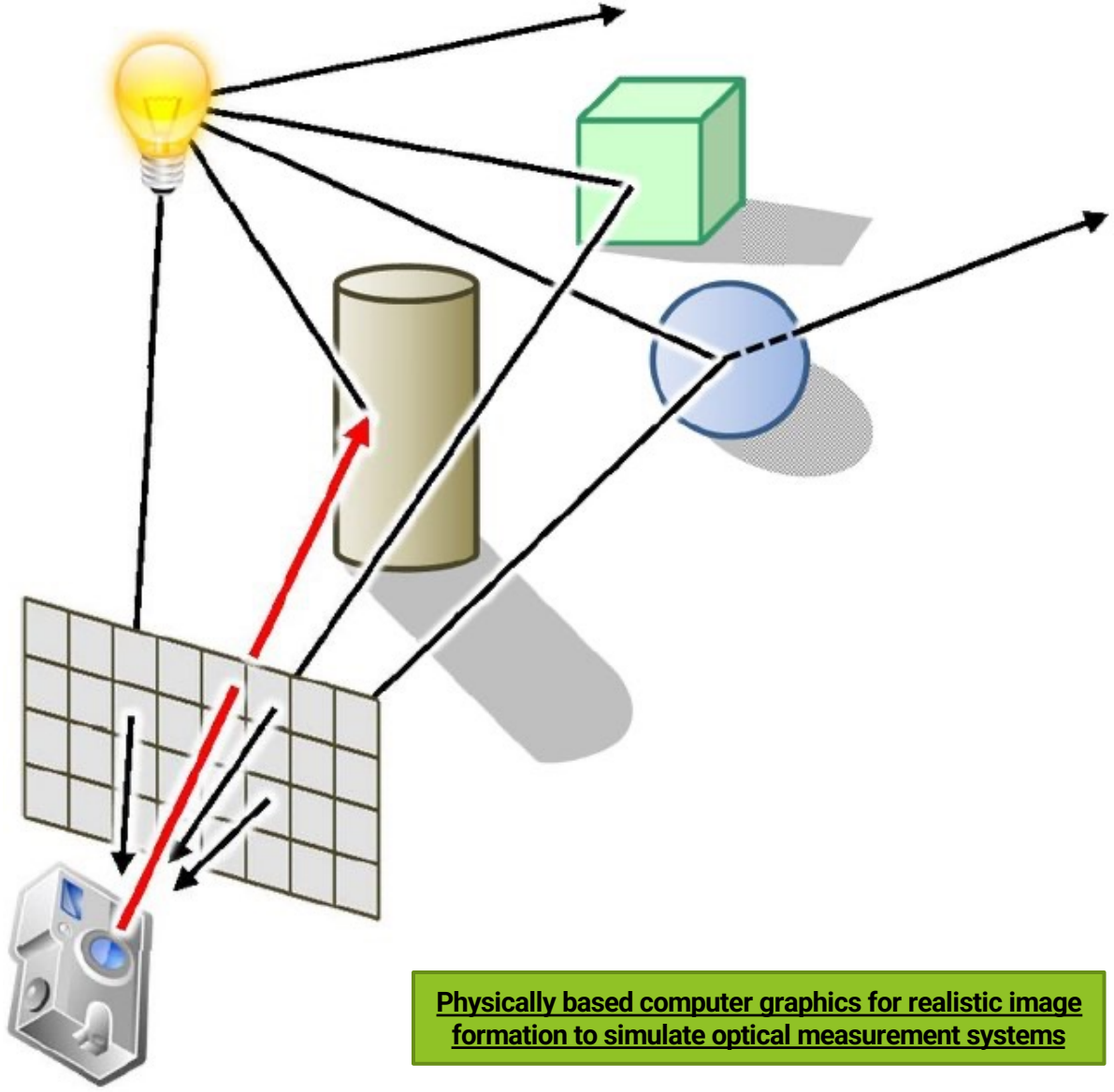
Easily parallelisable with respect to pixels and rays
=> Suitable for GPU

4D Optics for tracing light paths

- Object : codim 1
- Light : codim 1
- Camera : 2D screen + origin
- Ray : dim 1

$=>$ dim(Ray∩Object)=0

There are choices
Ex: Light “plane” generated by the 1D origin and a pixel on the 2D screen that intersects with 2D objects

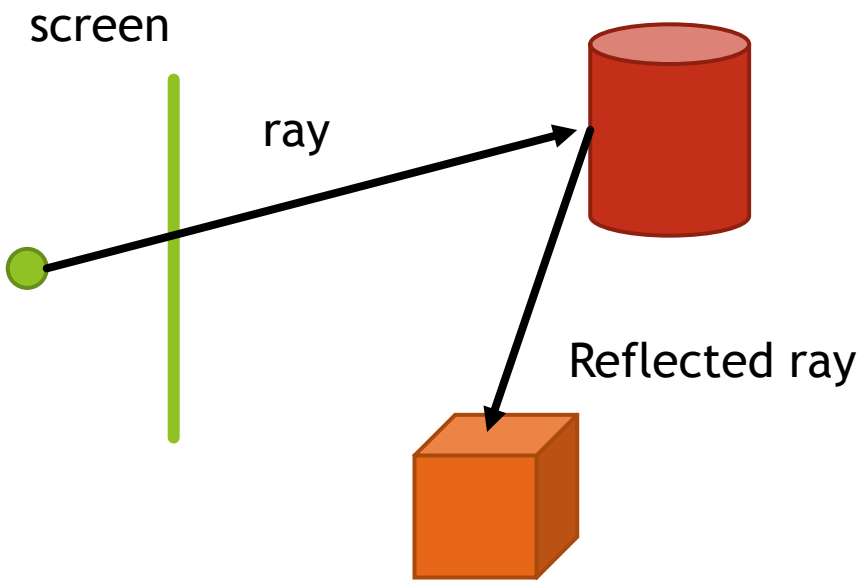


Physically based computer graphics for realistic image formation to simulate optical measurement systems

Codim 2 screen

- The codim 2 screen is the only non-trivial choice
- Let’s understand this by the analogy in 3D (Flatland analogy!)

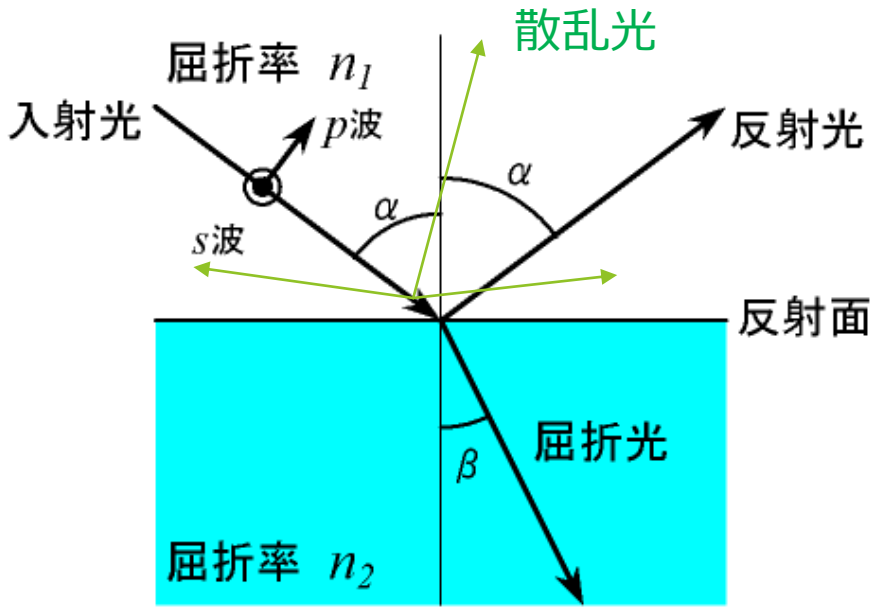
高次元のことを理解するには，“Flatland”式に次元を下げてイメージすると良い

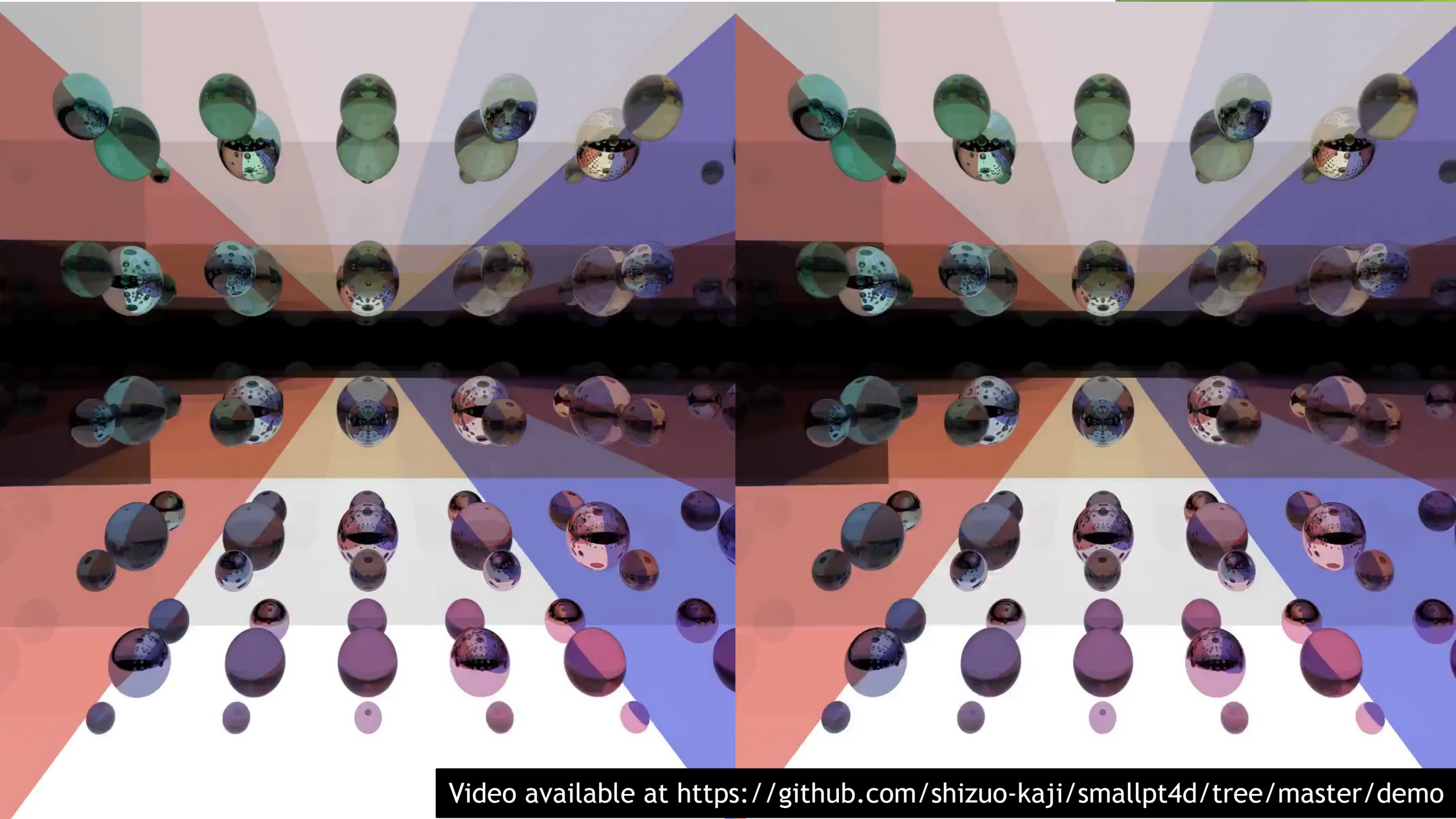


光線は最初，カメラ(線分：スクリーン, 視点；点)で張られる２次元空間(cone)を進む.
A ray travels within the 2D cone spanned by the origin and the screen until it hits something.
物体に当たって反射・屈折・散乱するときには，その２次元 cone から外れる.
=> 反射光が高次元の情報を capture
Reflected rays leave the 2D cone and capture the information of the other dimension

smallpt4d

- https://github.com/shizuo-kaji/smallpt4d
- In the current setting,
  reflection and refraction take place in the 2D subspace spanned by the incoming ray and the normal
  ＝＞ straightforward generalisation of 3D optics
- Diffusion is just random and straightforward




Video available at https://github.com/shizuo-kaji/smallpt4d/tree/master/demo

光学”代数”
”Optical algebra”

- 反射や屈折は，アフィン部分空間の間の演算とみなせる
  Reflection/Refraction can be seem as algebraic operations on affine subspaces