

Microsoft XNAを用いた カーボンナノチューブの立体構造3Dビューアの作成

September 2, 2011 (Fry.)
実験教育支援センター 茂木隆太

- Contents -

① 経緯

② 知識

③ 設計

④ 結果

⑤ 総括

⑥ 謝辞





カーボンナノチューブの立体的な構造図
を学会用ポスターに載せたいんだけど、
なんかいいアイディアないかな？

なに言っているの
かよくわからない
けど勉強して作る
か・・・。

著作権フリーで
・ ZigZag型
・ ArmChair型
・ Chiral型
が判別できるやつ



かつて・・・

平成19年度 第19回理工学部技術系職員研修発表 『初歩の3DCG学習会』



© Sumitaka Hase

知識



カーボンナノチューブ（詳細）

ログインまたはアカウント作成

- 炭素によって作られる六員環ネットワーク(グラフェンシート)が単層あるいは多層の同軸環状になった物質のこと。

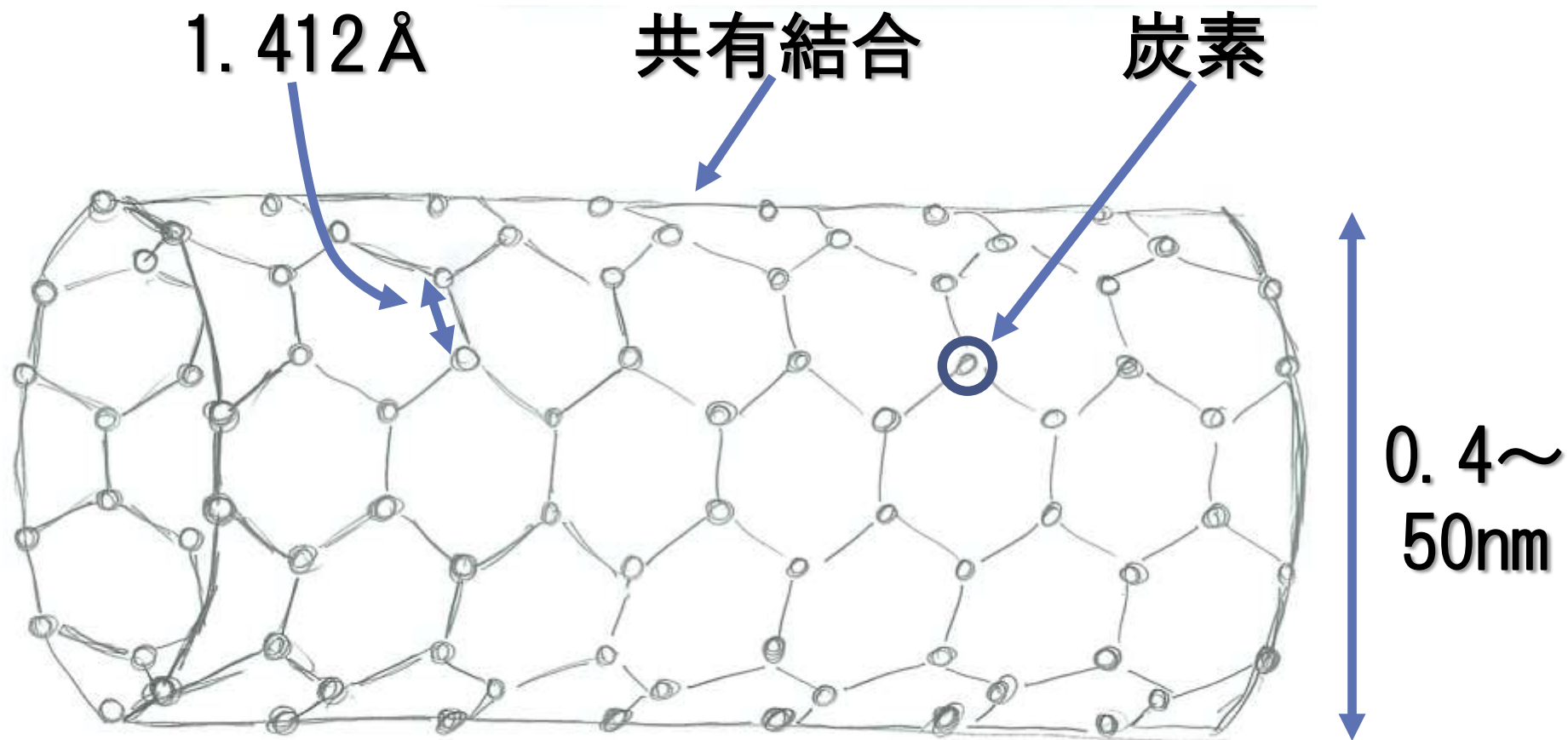
- グラフェンシートの幾何学的構造の違いによって、3種類のカーボンナノチューブが成立する。

幾何学的構造の全ては **カイラル指数(螺旋度)** によって決まる。

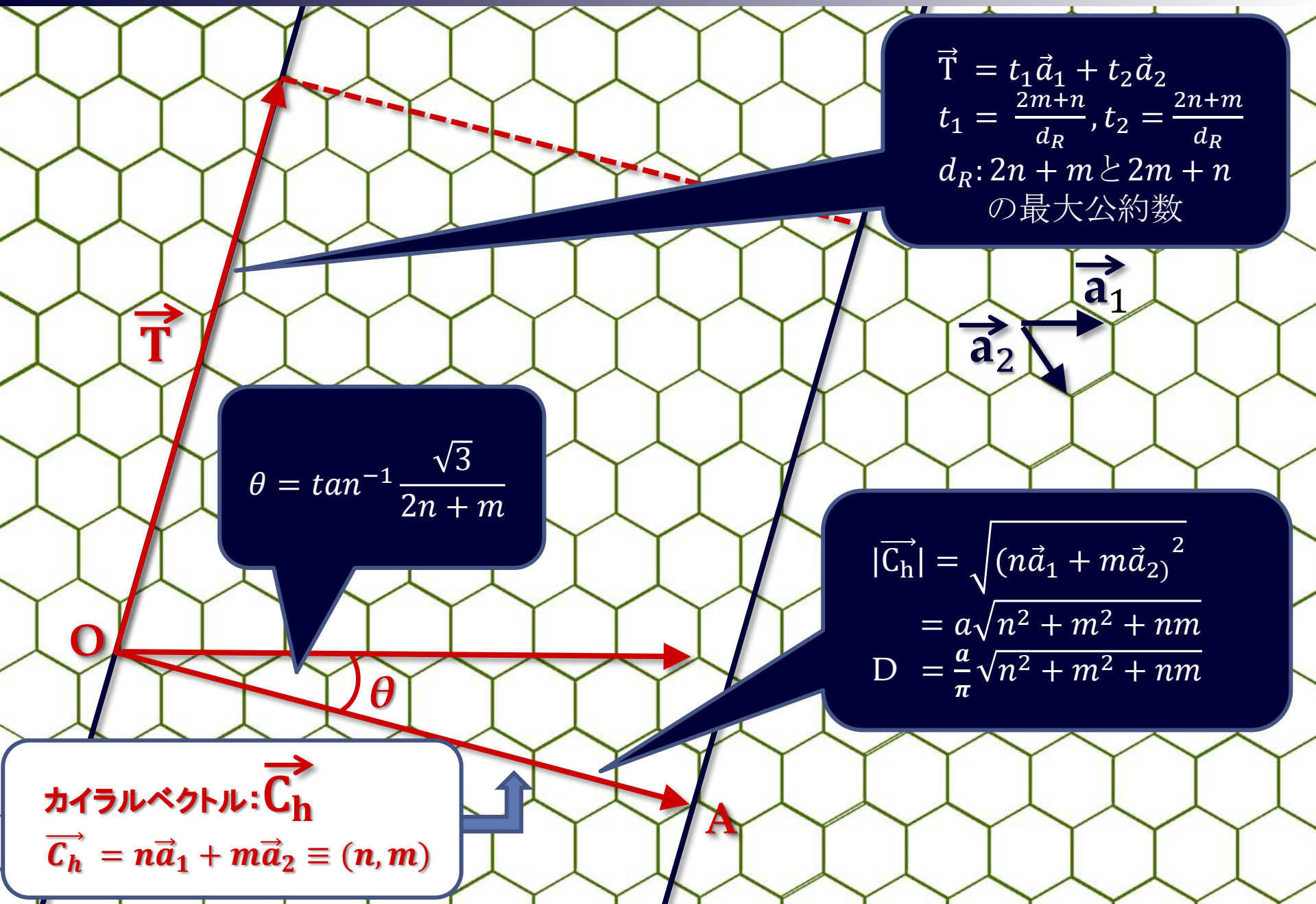


電子構造が変化（半導体？金属？）

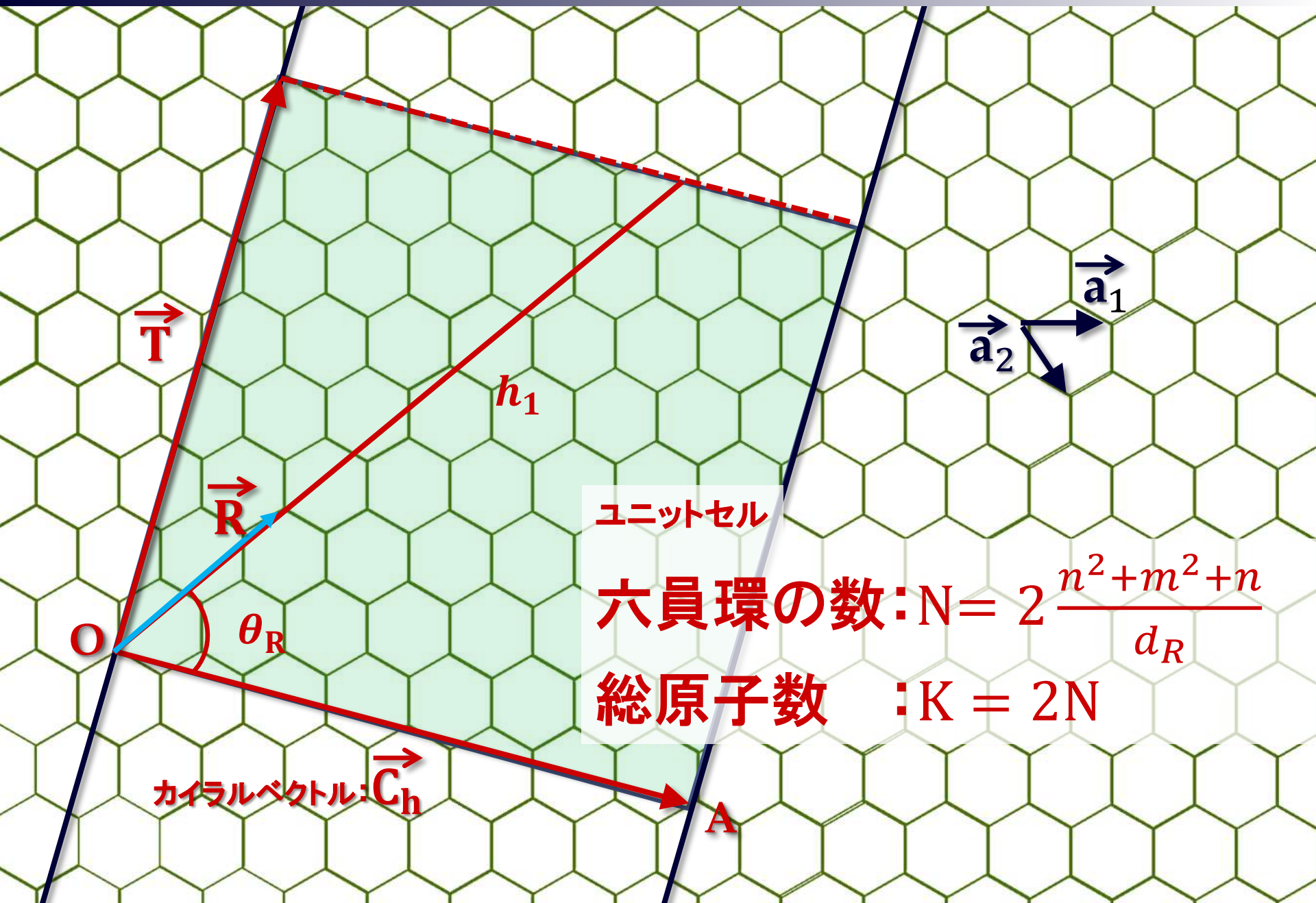
カーボンナノチューブ (構造図イメージ)



グラフエンシート (展開図)



グラフェンシート (展開図)



ユニットセル

$$\text{六員環の数: } N = 2 \frac{n^2 + m^2 + n}{d_R}$$

$$\text{総原子数 : } K = 2N$$

カイラルベクトル: \vec{C}_h

原子の座標

なんだかんだで、カイラル指数 (n, m) が決まれば・・・

α 原子座標

$$\begin{cases} x = r \cdot \cos \left(k \cdot \frac{2\pi |\vec{R}| \cos \theta_R}{|\vec{C}_h|} \right) \\ y = r \cdot \sin \left(k \cdot \frac{2\pi |\vec{R}| \cos \theta_R}{|\vec{C}_h|} \right) \\ z = (k|\vec{R}| - k_2 h_1) \cdot \sin \theta_R \end{cases}$$

カイラリティ

$$\vec{C}_h = n\vec{a}_1 + m\vec{a}_2 \equiv (n, m)$$

直径

$$D = \frac{a}{\pi} \sqrt{n^2 + m^2 + nm}$$

総六員環数

$$N = 2 \frac{n^2 + m^2 + n}{d_R}$$

総原子数

$$K = 2N$$

β 原子座標

$$\begin{cases} x = r \cdot \cos \left(k \cdot \frac{2\pi |\vec{R}| \cos \theta_R}{|\vec{C}_h|} - \frac{2\pi |d_2|}{|\vec{C}_h|} \right) \\ y = r \cdot \sin \left(k \cdot \frac{2\pi |\vec{R}| \cos \theta_R}{|\vec{C}_h|} - \frac{2\pi |d_2|}{|\vec{C}_h|} \right) \\ z = (k|\vec{R}| - k_2 h_1) \cdot \sin \theta_R + h_2 \end{cases}$$

但し

$$\theta_R = \tan^{-1} \frac{\sqrt{3}m}{2n+m} - \tan^{-1} \frac{\sqrt{3}q}{2p+q}$$

$$d_2 = a_{c-c} \cos \left(\theta + \frac{\pi}{6} \right)$$

$$h_2 = -a_{c-c} \sin \left(\theta + \frac{\pi}{6} \right)$$

$$k = 1, \dots, N$$

$$k_2 = \text{floor} \left(\frac{k|\vec{R}|}{h_1} \right)$$

つまり何が言いたいかというと・・・

カイラル指数(n, m)を最初に定義するだけで・・・

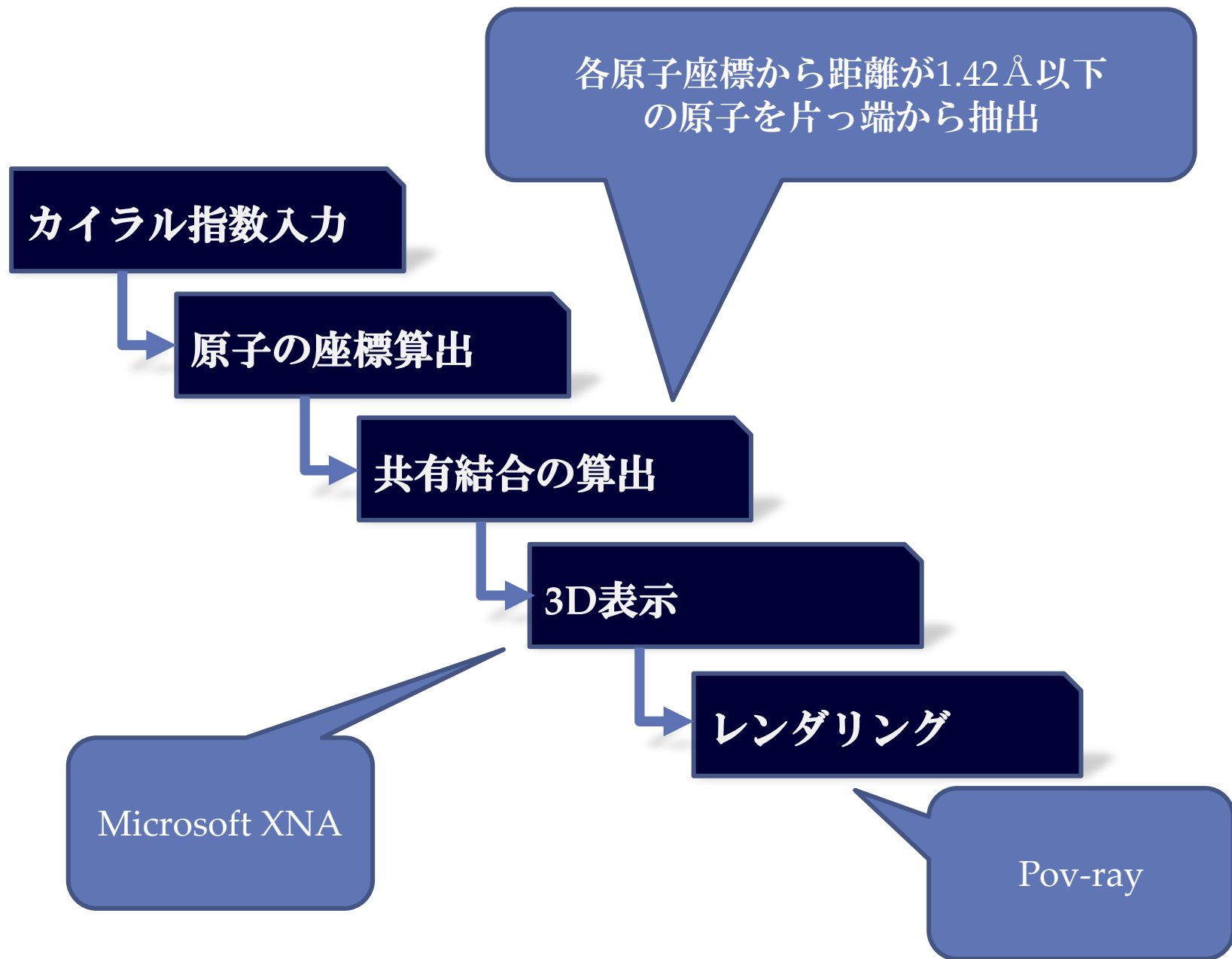
3次元上の原子の座標も、総原子数も、総六員環数も、直径も、電子状態(バンドギャップ)もすべて決まる。

ちなみに、 $n=m$ をArmChair型、 $m=0$ をZigZag型、その他をChiral型という

建設費

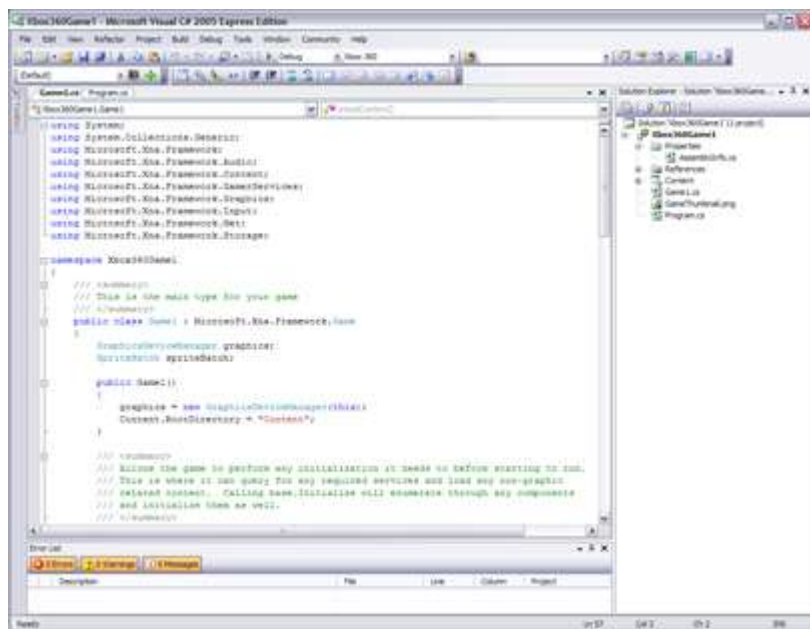


ソフトウェア設計



Microsoft XNAとは

- マイクロソフトによるコンピュータゲームデザイン、開発および管理を促進することを目的としたゲーム開発のためのAPI（フレームワーク）
- 要約するとDirectXシリーズの後継で、.NET frameworkに対応させたもの
- プログラム言語はC#

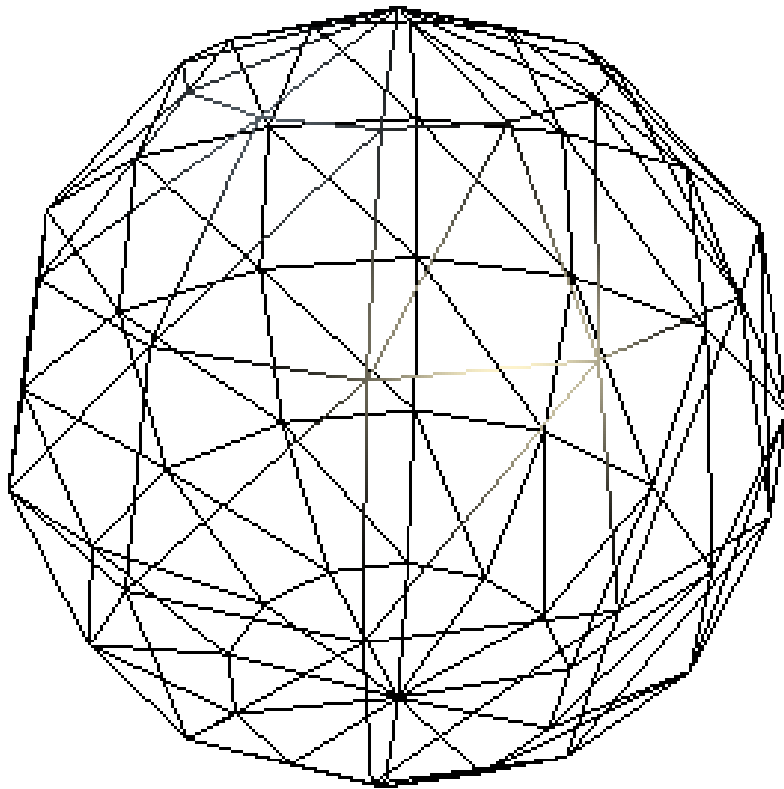


OpenGLとの違い

	OpenGL	Microsoft XNA
対応OS	Windows, Mac, Linuxなど	Windows専用
目的	CAD、デザインなど	ゲーム、エンターテインメントなど
機能	グラフィックスのみ	グラフィックス、サウンド、ネットワーク通信など
最適化されたビデオカード	NVIDIA 『Quadro』シリーズなど	NVIDIA 『GeForce』シリーズなど
その他	PlayStation3で採用	XBOX360で採用

3D表示：三次元で球を描くには

引数：直径、細かさ



```
int verticalSegments = tessellation;
int horizontalSegments = tessellation * 2;

float radius = diameter / 2;

// Start with a single vertex at the bottom of the sphere.
AddVertex(Vector3.Down * radius, Vector3.Down);

// Create rings of vertices at progressively higher latitudes.
for (int i = 0; i < verticalSegments - 1; i++)
{
    float latitude = ((i + 1) * MathHelper.Pi /
        verticalSegments) - MathHelper.PiOver2;

    float dy = (float)Math.Sin(latitude);
    float dxz = (float)Math.Cos(latitude);

    // Create a single ring of vertices at this latitude.
    for (int j = 0; j < horizontalSegments; j++)
    {
        float longitude = j * MathHelper.TwoPi / horizontalSegments;

        float dx = (float)Math.Cos(longitude) * dxz;
        float dz = (float)Math.Sin(longitude) * dxz;

        Vector3 normal = new Vector3(dx, dy, dz);

        AddVertex(normal * radius, normal);
    }
}

// Finish with a single vertex at the top of the sphere.
AddVertex(Vector3.Up * radius, Vector3.Up);

// Create a fan connecting the bottom vertex to the bottom latitude ring.
for (int i = 0; i < horizontalSegments; i++)
{
    AddIndex(0);
    AddIndex(1 + (i + 1) % horizontalSegments);
    AddIndex(1 + i);
}

// Fill the sphere body with triangles joining each pair of latitude rings.
for (int i = 0; i < verticalSegments - 2; i++)
{
    for (int j = 0; j < horizontalSegments; j++)
    {
        int nextI = i + 1;
        int nextJ = (j + 1) % horizontalSegments;

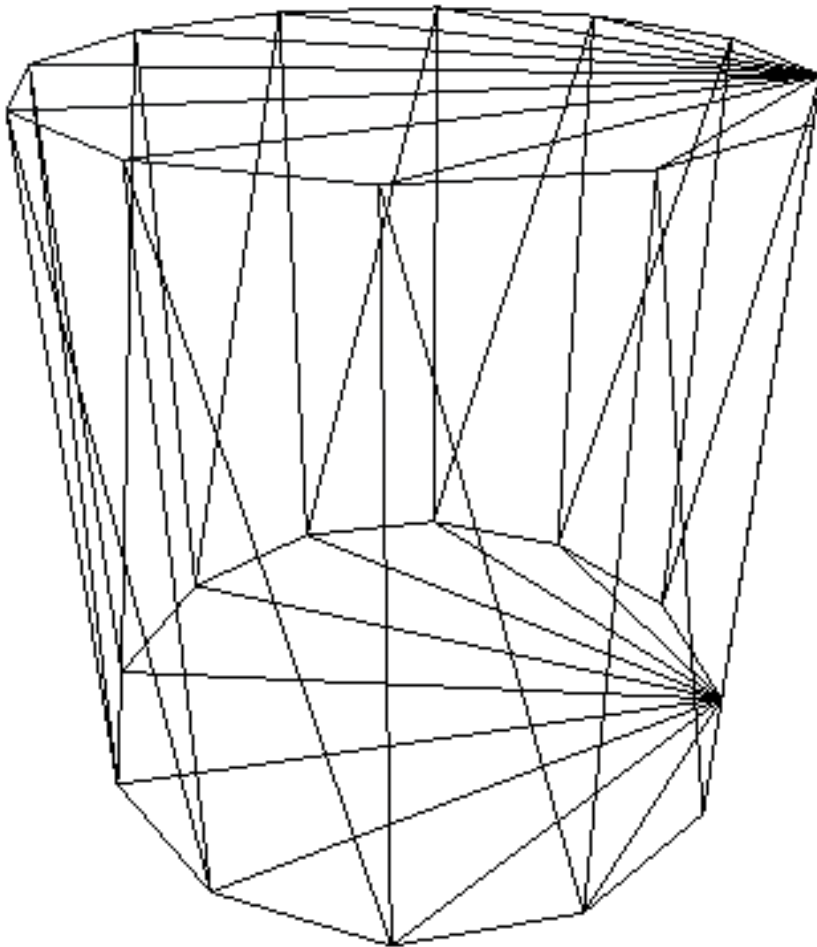
        AddIndex(1 + i * horizontalSegments + j);
        AddIndex(1 + i * horizontalSegments + nextJ);
        AddIndex(1 + nextI * horizontalSegments + j);

        AddIndex(1 + i * horizontalSegments + nextJ);
        AddIndex(1 + nextI * horizontalSegments + nextJ);
        AddIndex(1 + nextI * horizontalSegments + j);
    }
}

// Create a fan connecting the top vertex to the top latitude ring.
for (int i = 0; i < horizontalSegments; i++)
{
    AddIndex(CurrentVertex - 1);
    AddIndex(CurrentVertex - 2 - (i + 1) % horizontalSegments);
    AddIndex(CurrentVertex - 2 - i);
}
```


3D表示：三次元で円柱を描くには

引数：高さ、直径、細かさ



```
// new argument to change tessellation //
height /= 2;

float radius = diameter / 2;

// Create a ring of triangles around the outside of the cylinder.
for (int i = 0; i < tessellation; i++)
{
    Vector3 normal = GetCircleVector(i, tessellation);

    AddVertex(normal * radius + Vector3.Up * height, normal);
    AddVertex(normal * radius + Vector3.Down * height, normal);

    AddIndex(i * 2);
    AddIndex(i * 2 + 1);
    AddIndex((i * 2 + 2) % (tessellation * 2));

    AddIndex(i * 2 + 1);
    AddIndex((i * 2 + 3) % (tessellation * 2));
    AddIndex((i * 2 + 2) % (tessellation * 2));
}

// Create flat triangle fan caps to seal the top and bottom.
CreateCap(tessellation, height, radius, Vector3.Up);
CreateCap(tessellation, height, radius, Vector3.Down);

InitializePrimitive(graphicsDevice);
}

void CreateCap(int tessellation, float height, float radius, Vector3 normal)
{
    // Create cap indices.
    for (int i = 0; i < tessellation - 2; i++)
    {
        if (normal.Y > 0)
        {
            AddIndex(CurrentVertex);
            AddIndex(CurrentVertex + (i + 1) % tessellation);
            AddIndex(CurrentVertex + (i + 2) % tessellation);
        }
        else
        {
            AddIndex(CurrentVertex);
            AddIndex(CurrentVertex + (i + 2) % tessellation);
            AddIndex(CurrentVertex + (i + 1) % tessellation);
        }
    }

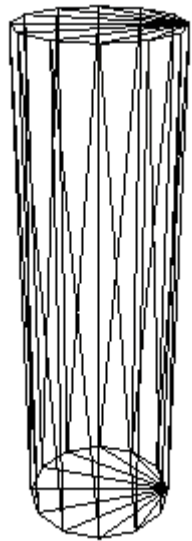
    // Create cap vertices.
    for (int i = 0; i < tessellation; i++)
    {
        Vector3 position = GetCircleVector(i, tessellation) * radius + normal * height;
        AddVertex(position, normal);
    }
}

static Vector3 GetCircleVector(int i, int tessellation)
{
    float angle = i * MathHelper.TwoPi / tessellation;

    float dx = (float)Math.Cos(angle);
    float dz = (float)Math.Sin(angle);

    return new Vector3(dx, 0, dz);
}
```

3D表示：円柱を任意の座標に回転させるには



X軸に対する回転軸

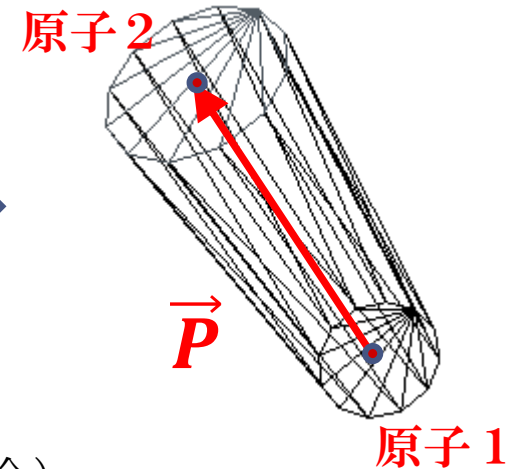
$$\alpha = \cos^{-1} \frac{P_y}{|\vec{P}|}$$



Y軸に対する回転軸

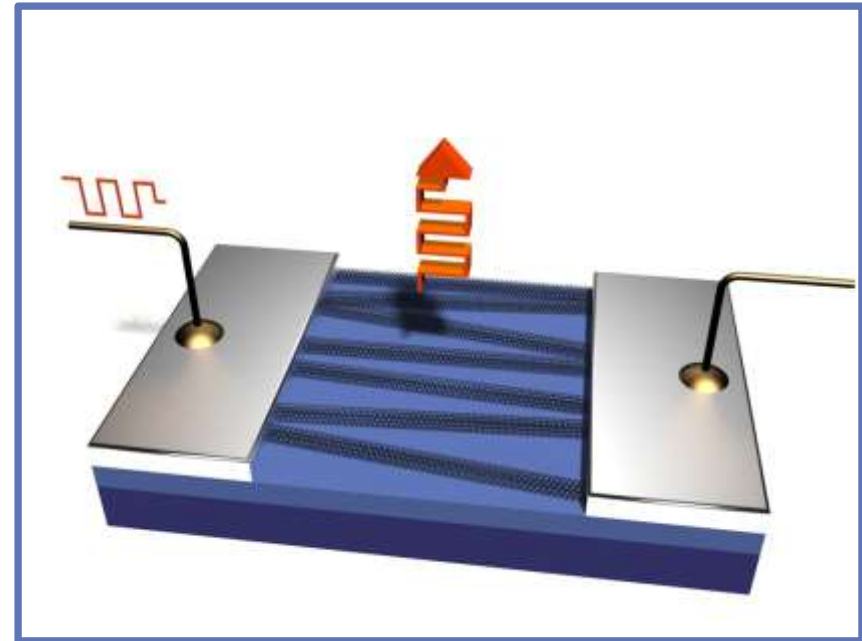
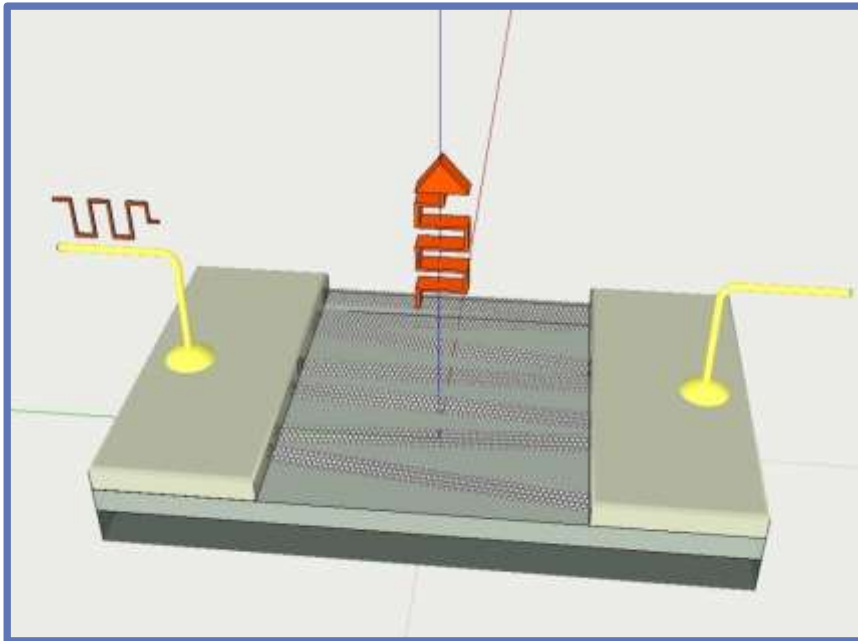
$$\begin{cases} \beta = \tan^{-1} \frac{P_x}{P_z} & (P_z > 0 \text{ の場合}) \\ \beta = \tan^{-1} \frac{P_x}{P_z} + \pi & (P_z \leq 0 \text{ の場合}) \end{cases}$$

但し、 $P_z = 0$ 且つ $P_x < 0$ なら $-\beta$



レンダリング（レイトレーシング）

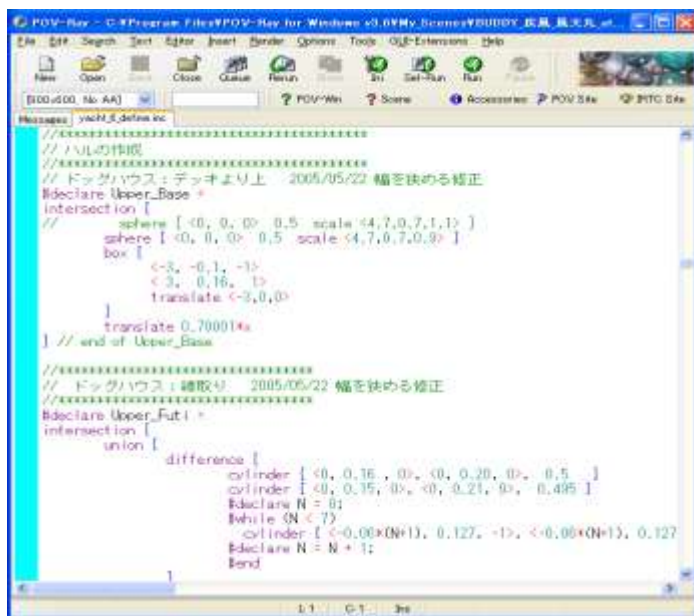
- 主に3次元空間内を伝わる光線（波）の伝播経路を追跡することで現実世界の物理現象を仮想的にシミュレートする計算手法



あら、きれい

Pov-rayとは

- 多くのコンピュータプラットフォームで利用出来るオープンソースのレイトレーシングソフトウェア（3Dレンダリングエンジン）
- マクロとループを搭載したチューリング完全のSDL (scene description language)で記述

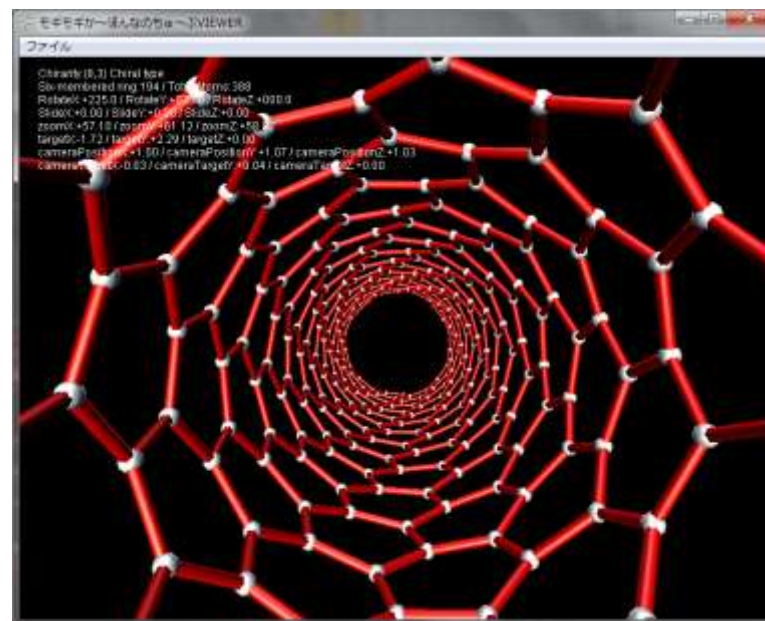


その他要求される機能

機能	操作方法
球、円柱のポリゴンの細かさの設定	テキストボックス
回転・移動・拡大縮小	マウス（左・右・中央）ドラッグ
球・円柱・背景の色・透過率の変更	スクロールバー（RGB・A）
球・円柱・全体の拡大縮小	スクロールバー（RGB・A）
各種効果 （ワイヤーフレーム） （FOG、アニメーション）	チェックボックス
Pov-ray用ファイル 簡易エディタ	テキストボックス、ボタン

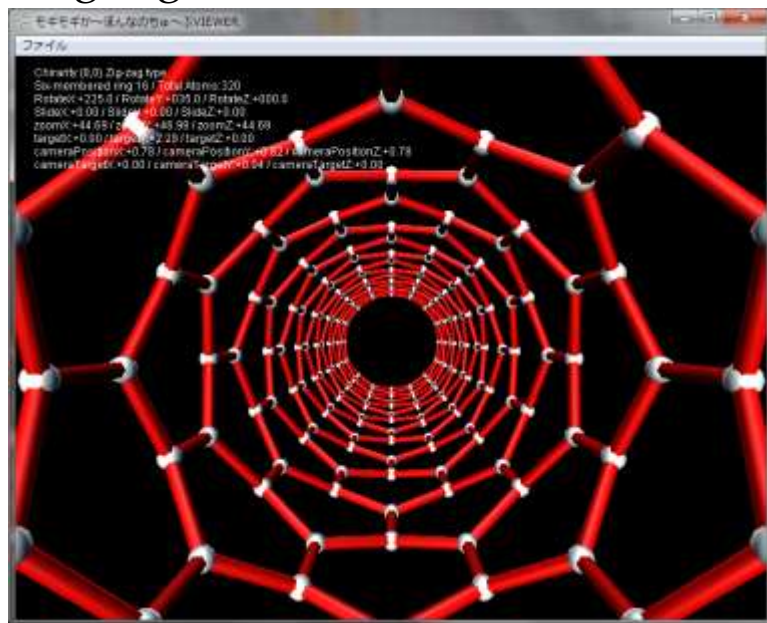


Chiral型 (8, 3)

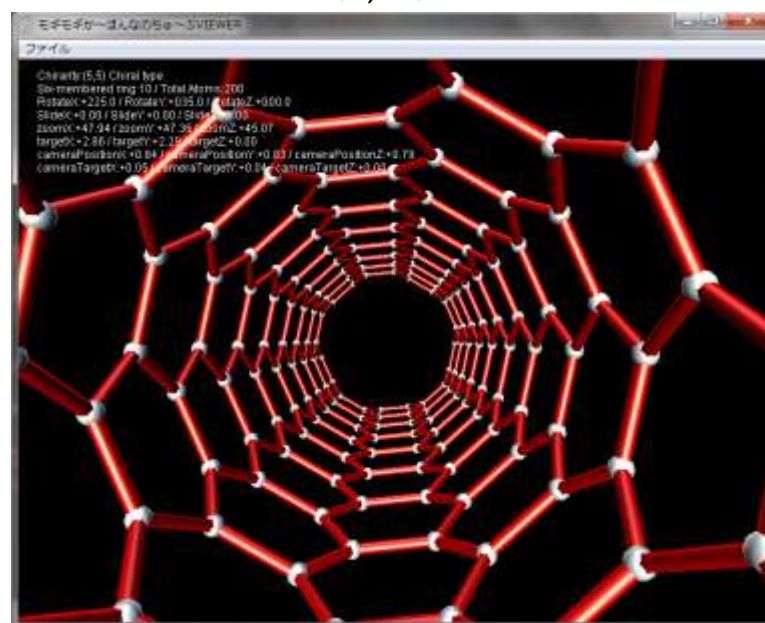


ビューア (Z軸方向)

ZigZag型 (8, 0)

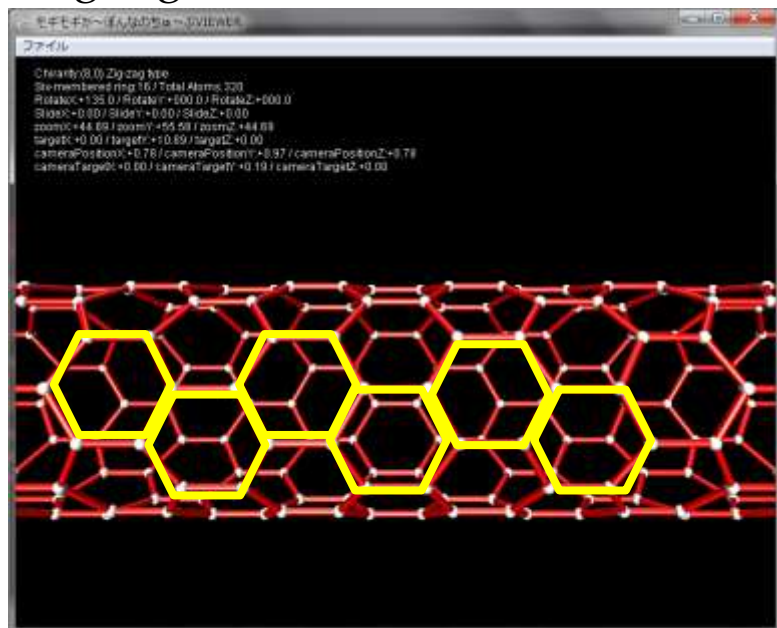


ArmChair型 (5, 5)

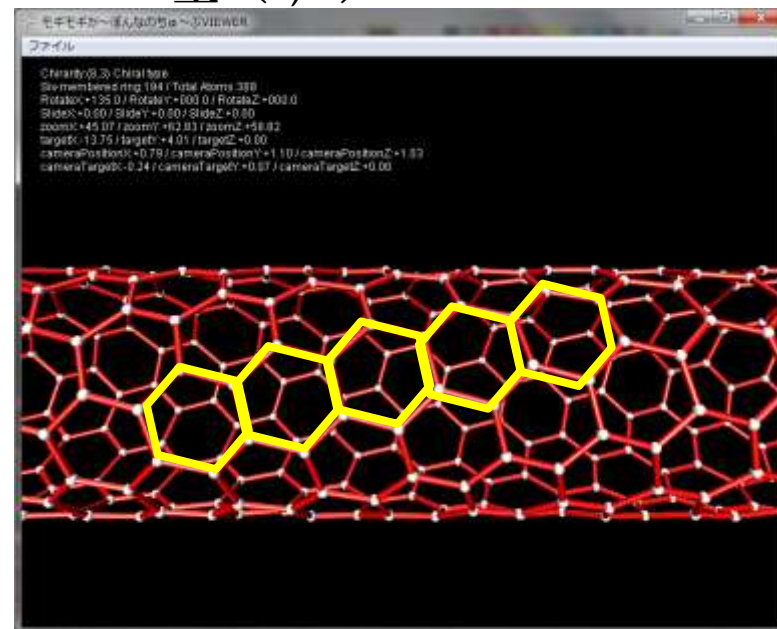


ビューア (X軸方向)

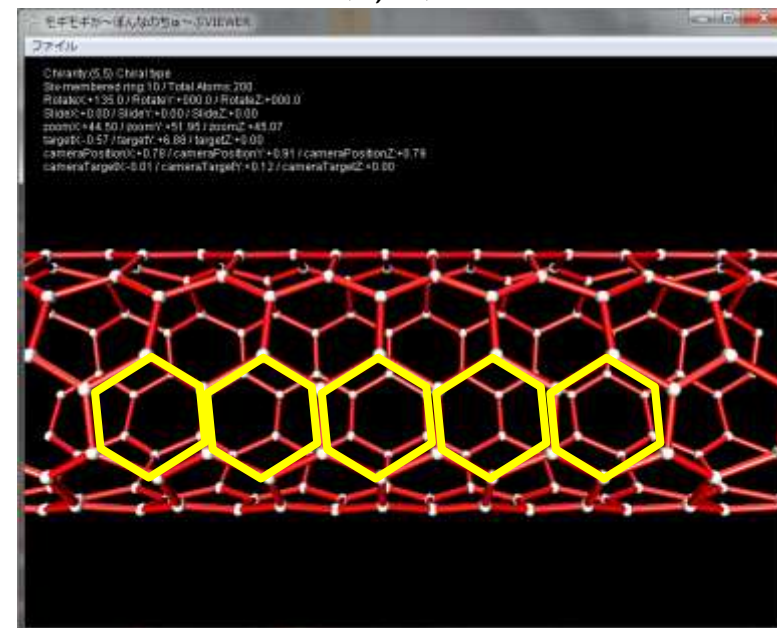
ZigZag型 (8, 0)

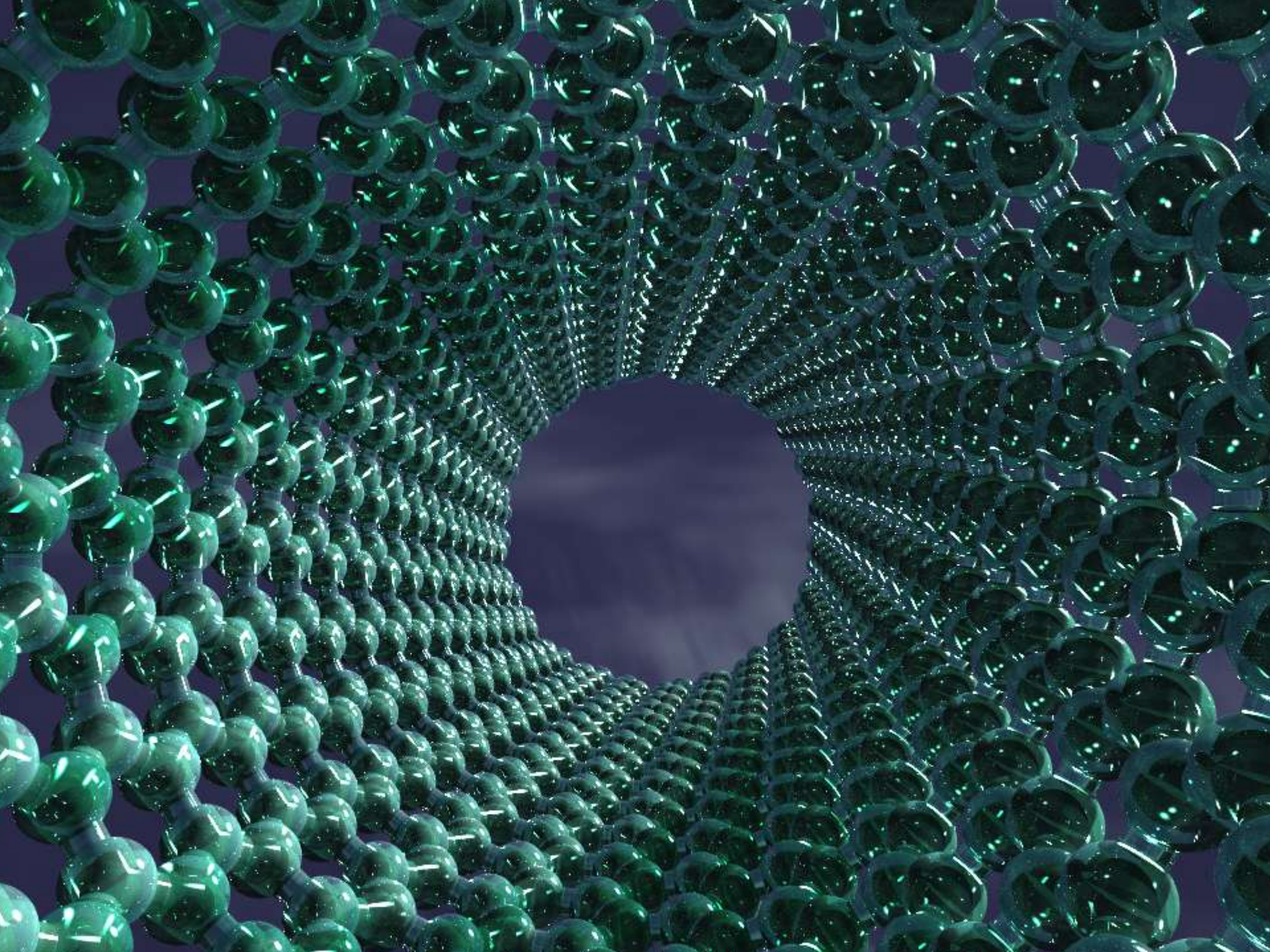


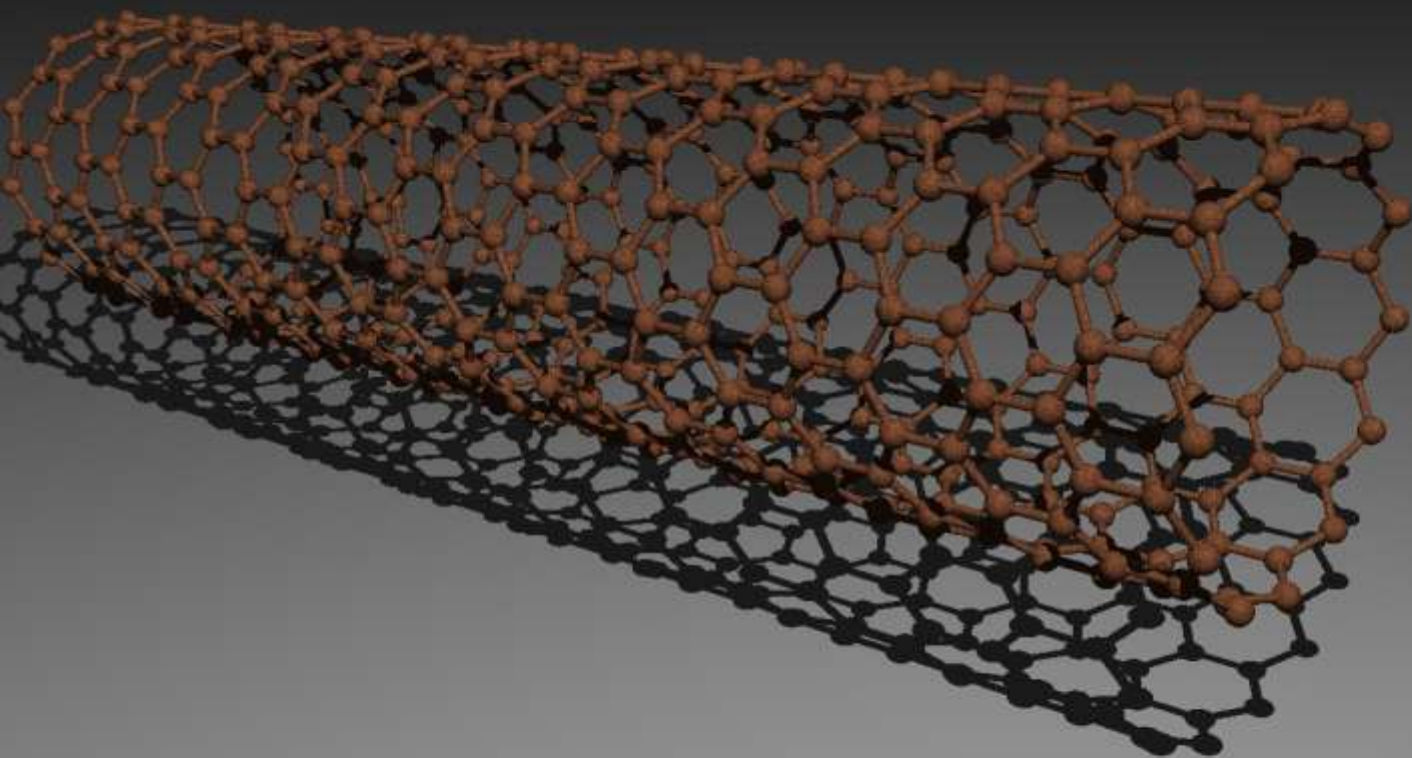
Chiral型 (8, 3)

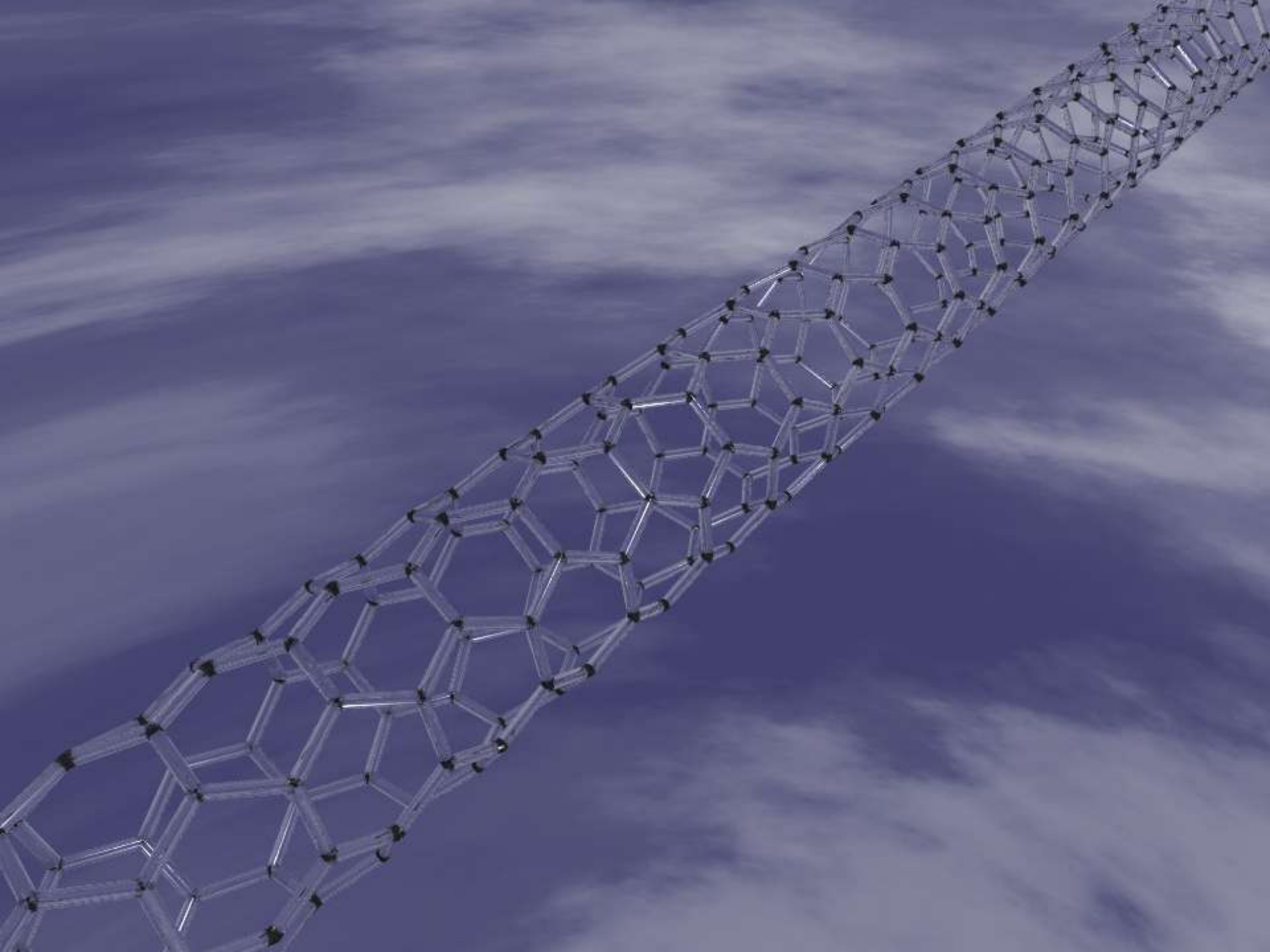


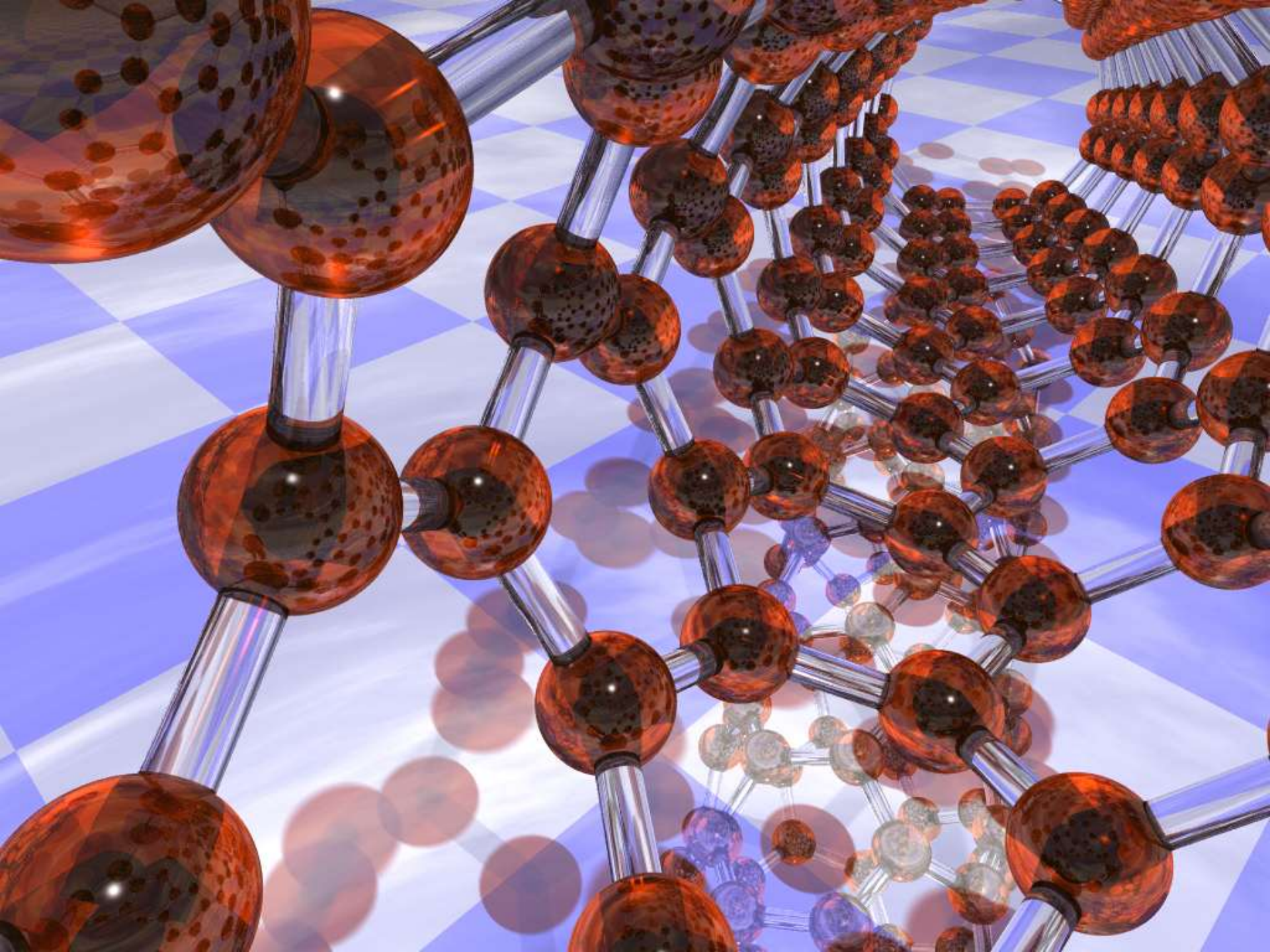
ArmChair型 (5, 5)

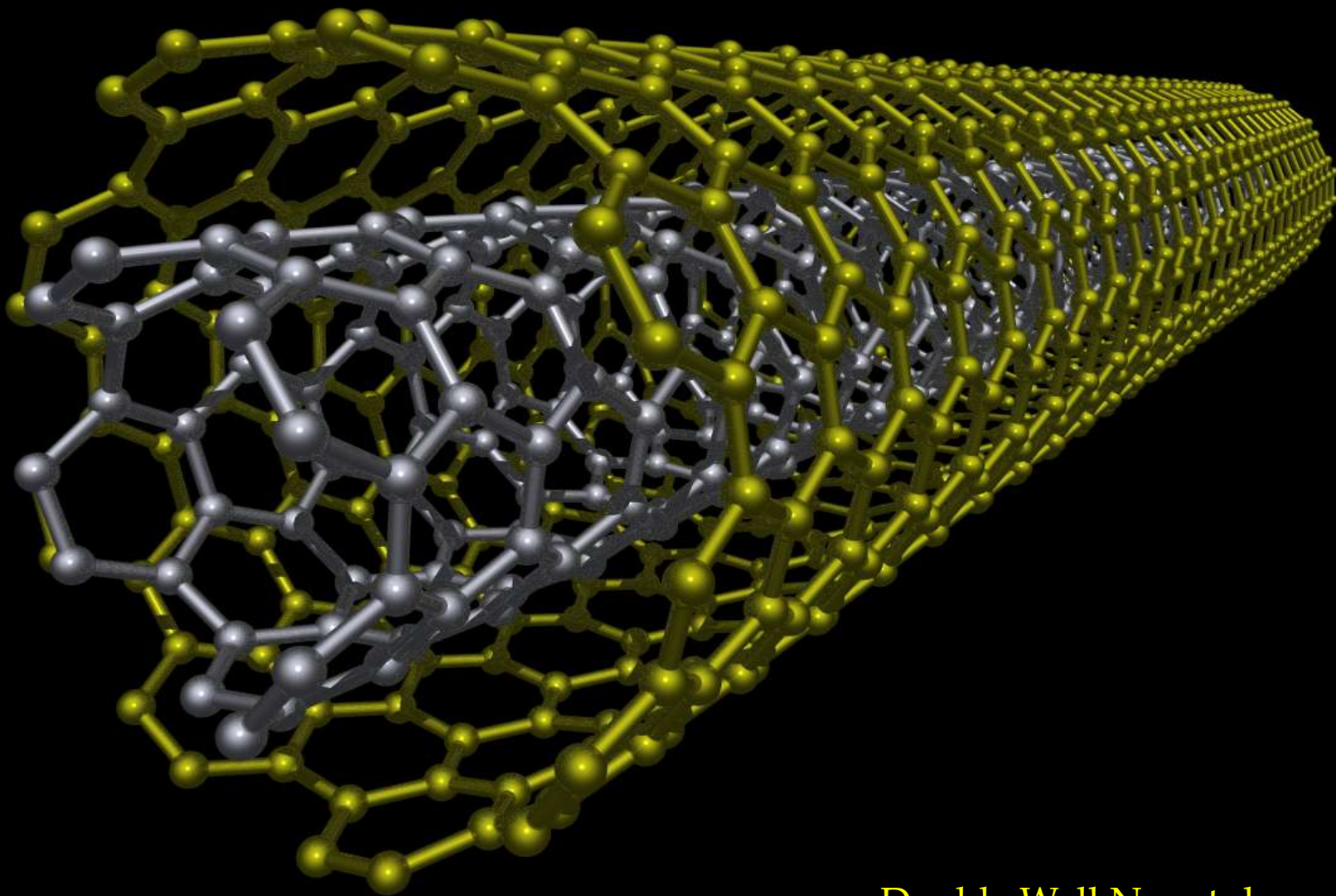




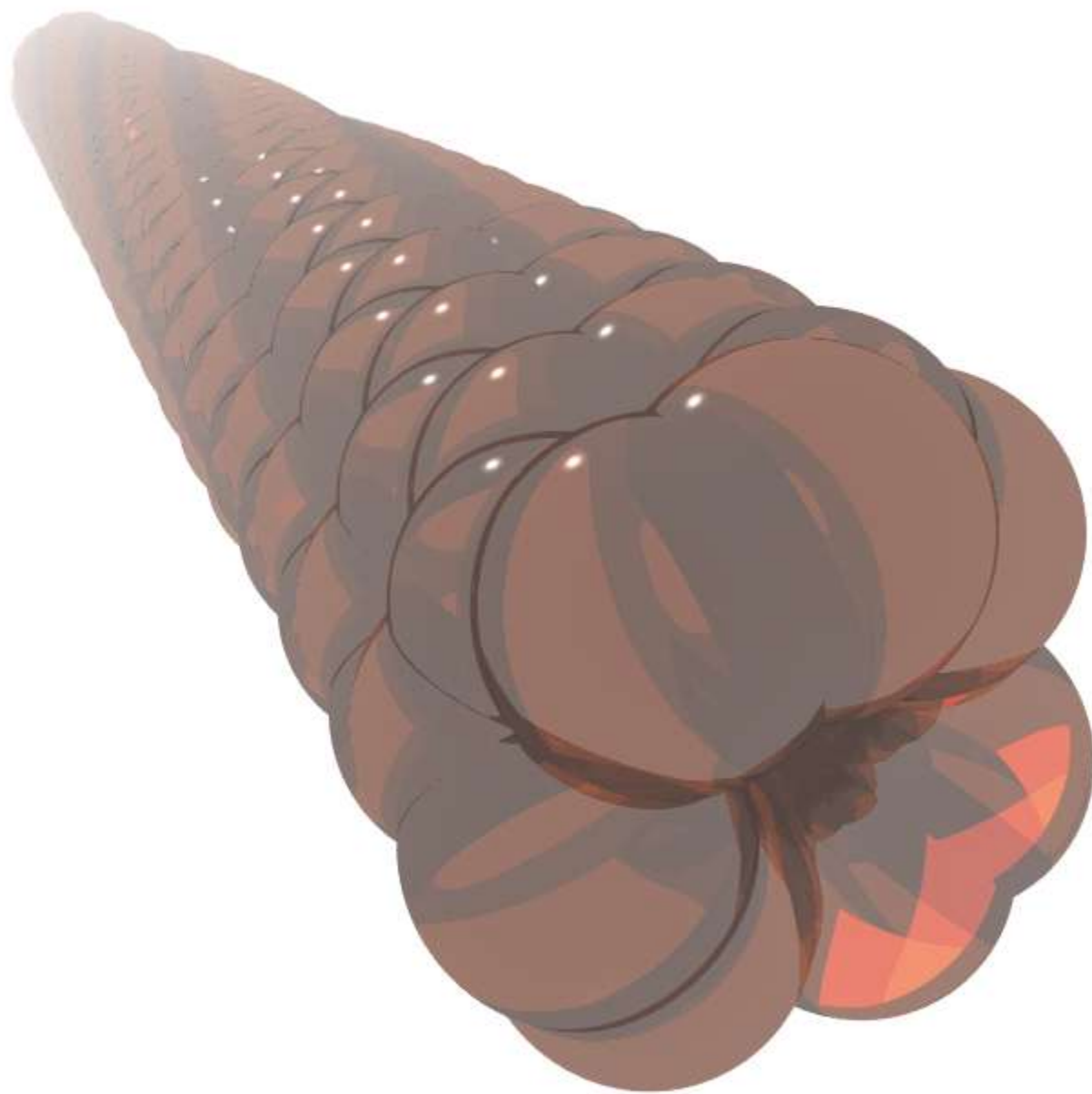








Double Wall Nanotube

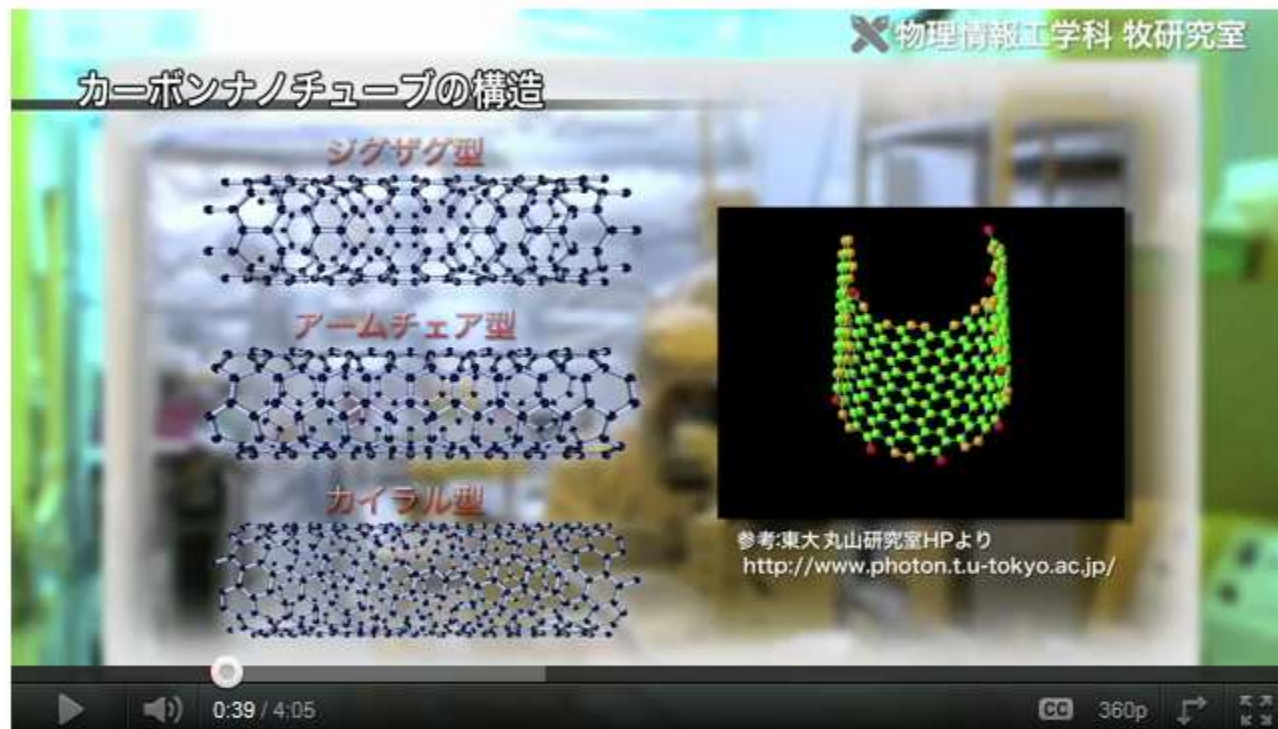


ナノ材料のデバイス化による新たな物性発現と機能デバイス開発

keiouniversity

464 件の動画

チャンネル登録



評価する 追加先 共有

1,209

keiouniversity さんが 2010/03/24 にアップロード

「ナノ材料のデバイス化による新たな物性発現と機能デバイス開発」

もっと見る

高評価 0 人、低評価 0 人

関連動画



光で世界を結ぶ:フォトニックネットワークの構築を目指して

ユーザー: keiouniversity
再生回数 1,099 回



カーボンナノチューブを使って量子オプトエレクトロニクスを目指して

ユーザー: keiouniversity
再生回数 1,124 回



塚田研究室:光・画像工学による腫瘍検出法の開発

ユーザー: keiouniversity
再生回数 732 回



量子ドット構造によるスピントロクス・デバイスで未来を拓く

ユーザー: keiouniversity
再生回数 2,746 回



インド タタ自動車 ナノ

ユーザー: namasuteindia
再生回数 9,320 回



未来研究会フォーラム 趣意説明

ユーザー: doyama2009
再生回数 193 回

新世紀

舞臺

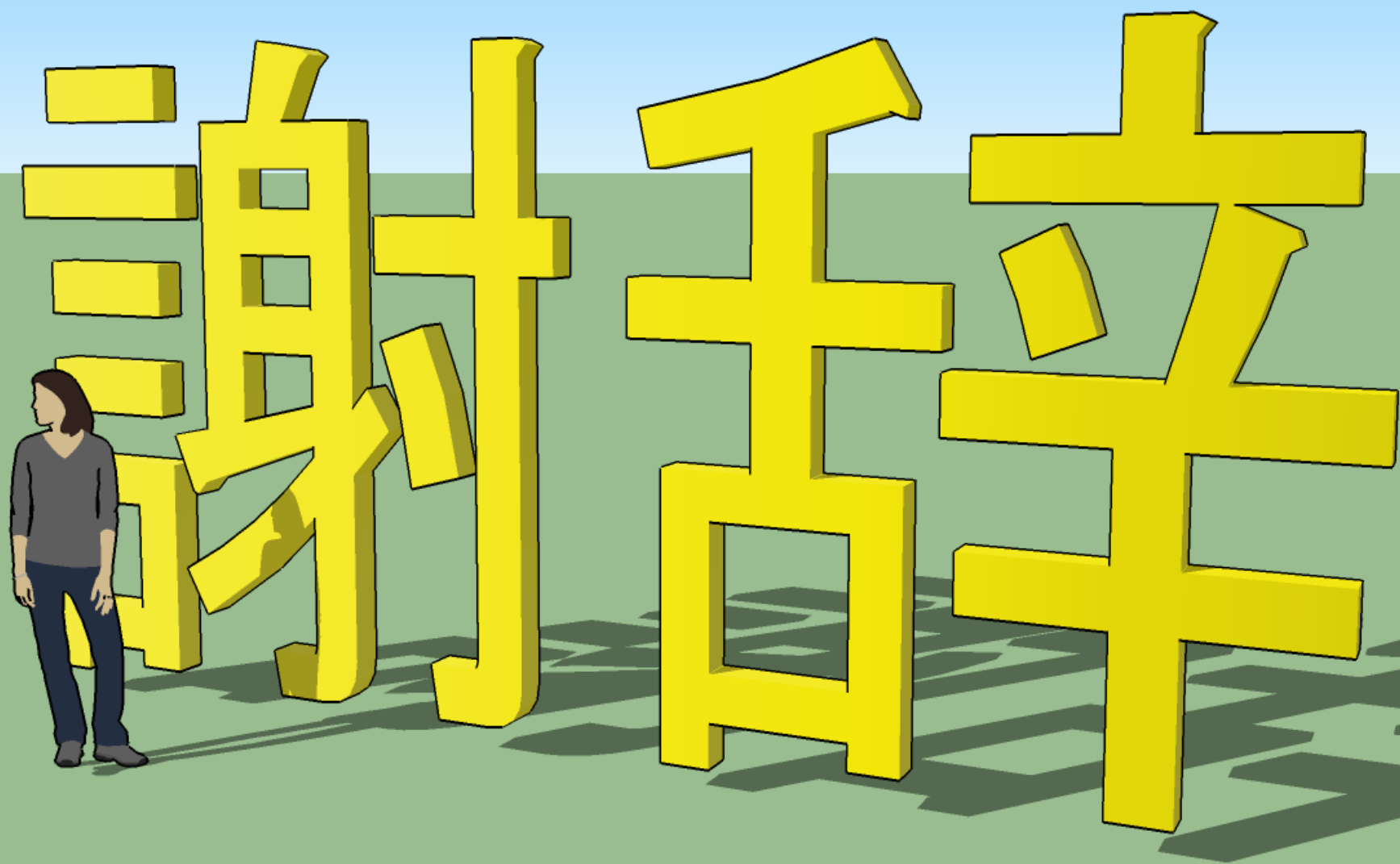


未解決問題

- ハットも計算で作れるのか？
⇒ 五員環の必要？
- カイラル指数の違うチューブの結合は？
⇒ 七員環が必要？
- 電子状態（エネルギーバンド）も表示する？
⇒ ブロッホ軌道を用いて、ユニットセル中での波数 k に対するエネルギー固有値 $E(k)$ として求めることができる？
- メモリが足りない？

まとめ

- 3次元描画ソフトを実際に作成することで、そのアルゴリズムや、設計方法が理解できた。
- カーボンナノチューブの構造が理解できた。
- 時代は（おそらく）3Dなので、今後色々な場面で応用していきたい。実験の資料など。



謝辞

- デバッグに付き合ってくださった物理情報工学科牧先生、牧研の皆さん
- 線形代数を指導してくだった物理情報工学科伊藤研の原田くん
- いろいろな我儘を聞いてくださった実験教育支援センター窪添さん
- 見ず知らずの私に座標算出プログラムを提供してくださいましたデラウェア大学のDr. Jeffrey T. Freyさん

Thank you very much

付録

3Dジオメトリパイプライン

