

# イメージ・アルファ画像処理ライブラリ 3DView マニュアル

第 1.00 版

株式会社 イメージ・アルファ

## はじめに

本ドキュメントは、イメージ・アルファ画像処理ライブラリ IALIB の 3DView について解説しています。

表 0-1 本ドキュメントの適用 IALIB バージョン

ソフト名	バージョン番号
IALIB	Version 1.2.2

## 目次

1. 機能.....	4
2. 使用方法.....	5
3. 関数.....	17
4. 列挙体、構造体.....	39
5. 補足.....	41

# 1. 機能

IALIB では、3次元コンピュータグラフィックス機能(3DView)を提供しています。  
3次元空間内の立体的な物体を、簡単に表示することができます。

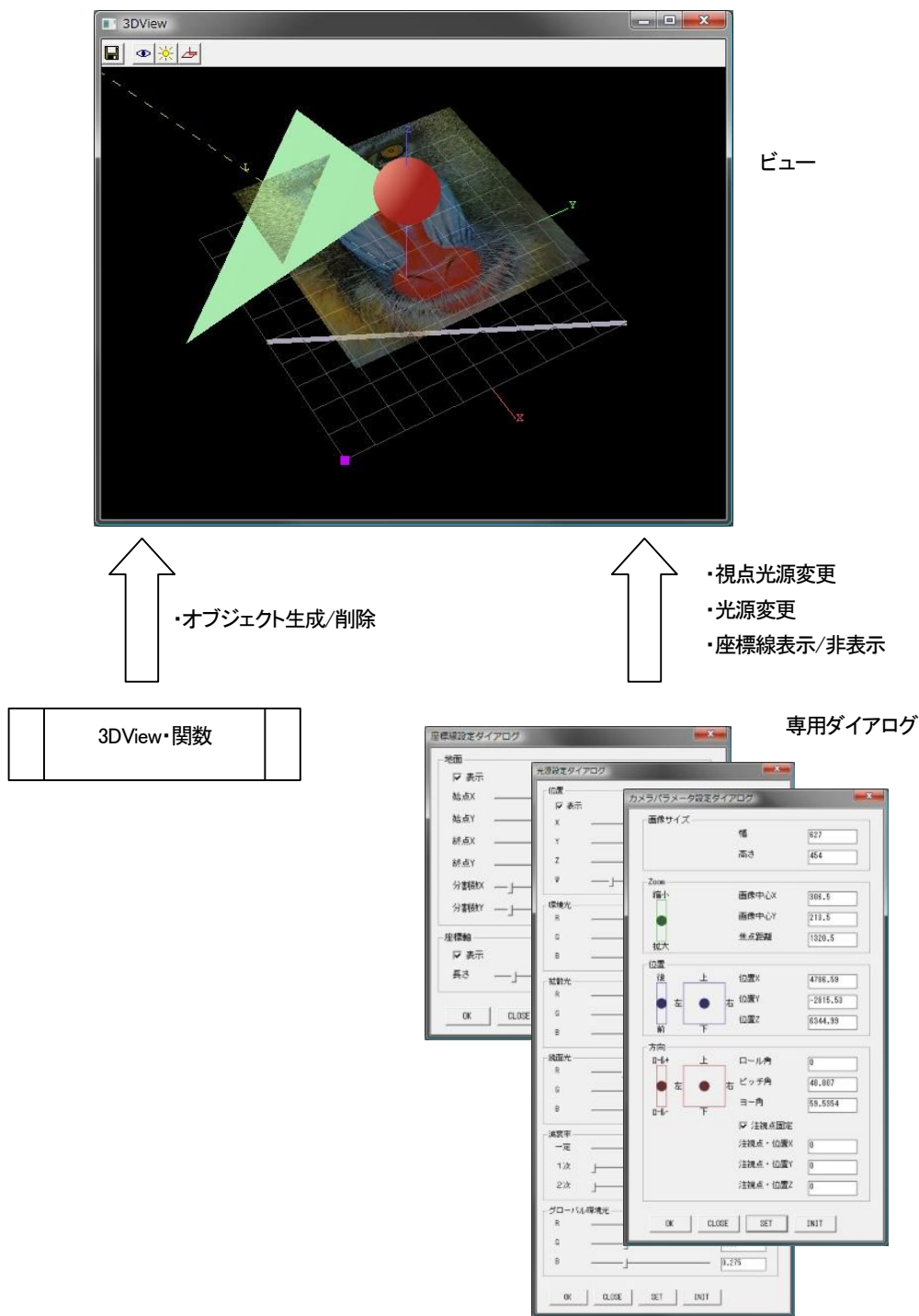


図 1-1 設定と表示

## 2. 使用方法

### 2.1. ソースコード

ソースコードでは、3DView の機能を有効化し、オブジェクトを生成する必要があります。

下のサンプルコードは、先ず赤色の点を表示し、次に緑色の四角形を表示するだけのアプリケーションです。実行すると図 2-1 のように表示されます。

```
int APIENTRY WinMain( HINSTANCE hInstance,
                    HINSTANCE hPrevInstance,
                    LPSTR lpCmdLine,
                    int nCmdShow )
{
    int viewID;
    char inifileName[] = "./inifile/test_3DView_sample.ini";

    int point_objID = -1;
    IA_3DV_Point point;
    float point_size;
    IA_3DV_Color point_color;
    int quad_objID = -1;
    IA_3DV_Point quad_point[4];
    IA_3DV_Material quad_material;

    // IALIB・各種有効化
    IA_Open();                                     → 2.1.1節

    // IALIB・3DView有効化
    IA_3DV_Open();                                 → 2.1.1節

    // IALIB・Eval有効化
    IA_Eval_Enable_MultiPort( 1, 1 );             → 2.1.1節

    // 3DView初期化
    viewID = IA_3DV_Create();                       → 2.1.2節

    // 制御パラメータ登録
    IA_3DV_RegisterParam( viewID );                → 2.1.5節

    // iniファイル読み込み
    IA_Eval_LoadIniFile( inifileName );           → 2.1.5節
}
```

```

// 点の生成
point.x = 1000.0F;    point.y = 1000.0F;    point.z = 0;
point_size    = 5;
point_color.r = 1.0F;  point_color.g = 0.0F;  point_color.b = 0.0F;  point_color.a = 1.0F;
point_objID = IA_3DV_CreatePoint( viewID, &point, point_size, &point_color, NULL );    → 2.1.3節

// 表示
IA_3DV_Show( viewID, 1 );    → 2.1.4節
MessageBox( NULL, "OKを押せば表¥示を終了します。", "表¥示(1回目)", MB_OK );

// 点の削除
IA_3DV_DeleteObject( viewID, point_objID );

// 四角形の生成
quad_point[0].x = -500.0F;    quad_point[0].y = -500.0F;    quad_point[0].z = 200.0F;
quad_point[1].x =  500.0F;    quad_point[1].y = -500.0F;    quad_point[1].z = 200.0F;
quad_point[2].x =  500.0F;    quad_point[2].y =  500.0F;    quad_point[2].z = 200.0F;
quad_point[3].x = -500.0F;    quad_point[3].y =  500.0F;    quad_point[3].z = 200.0F;
IA_3DV_GetInitMaterial( &quad_material );
quad_material.ambient.g = 1.0F;
quad_material.diffuse.g = 1.0F;
quad_objID = IA_3DV_CreateQuad( viewID, quad_point, &quad_material, NULL );    → 2.1.3節

// 表示(更新)
IA_3DV_Show( viewID, 1 );
MessageBox( NULL, "OKを押せば表¥示を終了します。", "表¥示(2回目)", MB_OK );    → 2.1.4節

// iniファイル書き込み
IA_Eval_SaveIniFile( inifileName );    → 2.1.5節

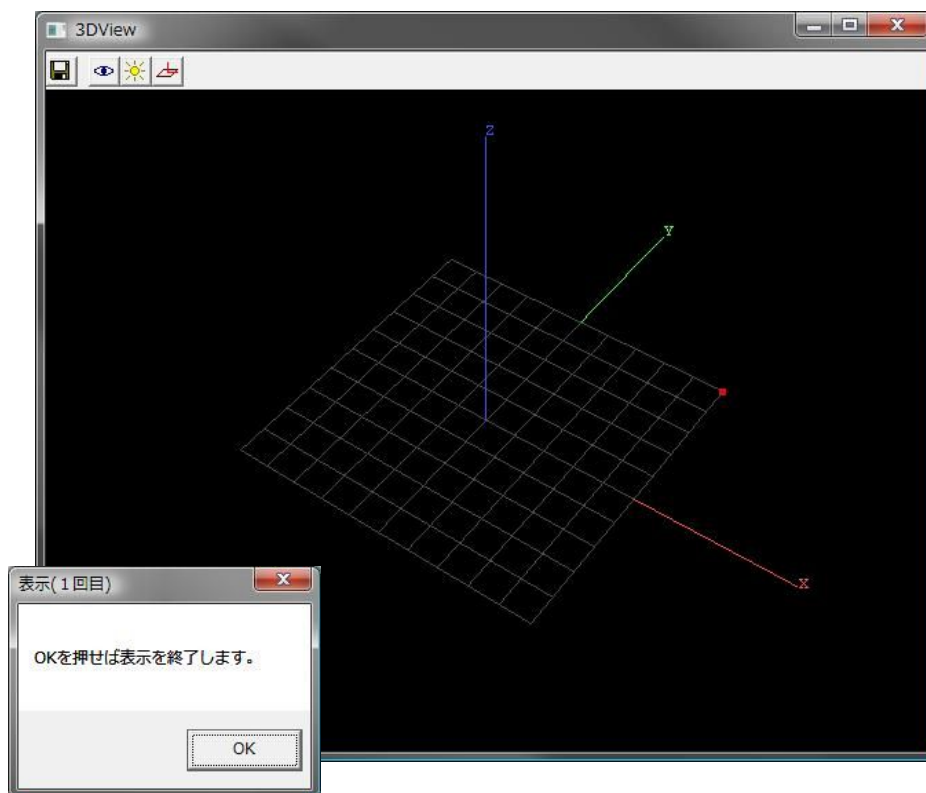
// IALIB・Eval無効化
IA_Eval_Disable();

// IALIB・3DView無効化
IA_3DV_Close();

// IALIB・無効化
IA_Close();

return 0;
}

```



↓ OK 押下

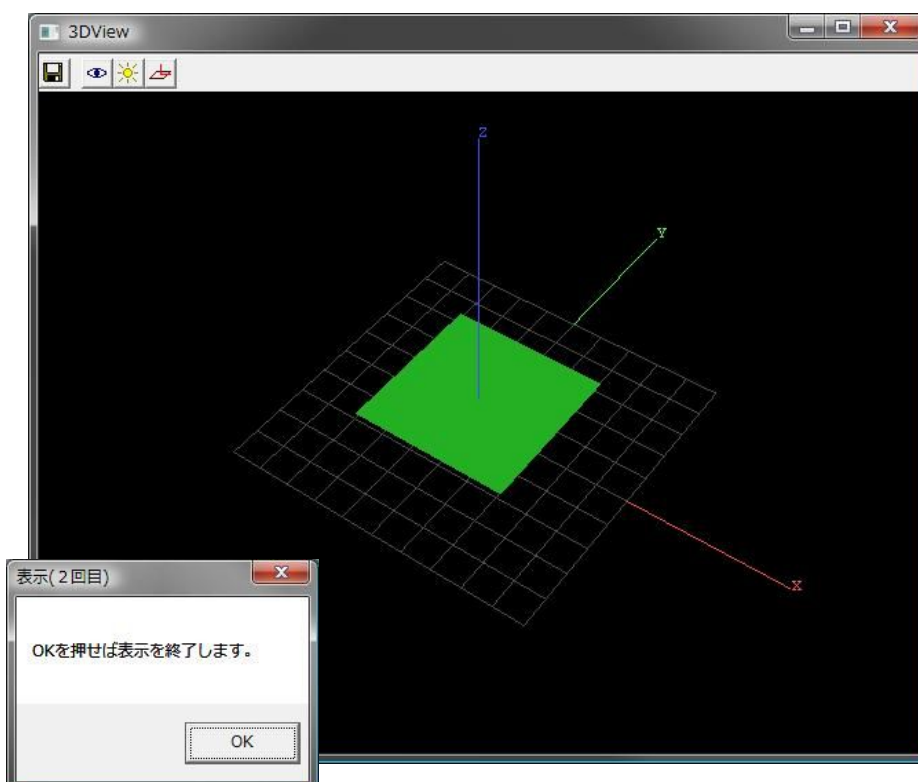


図 2-1 サンプルコードの表示

### 2.1.1. 各種有効化

IALIB全体の有効化、および3DViewの有効化を最初に行います。無効化されるまで機能が使用できます。

このサンプルコードではIALIBのEval機能を使用しています(2.1.5節)ので、Evalも有効化しています。

### 2.1.2. ビューの生成

関数 IA\_3DV\_Create によりビューを生成します。戻り値はビューIDです。  
オブジェクトの生成などはビュー単位で行います。

(ただし、現在ビューを複数生成することはできません。)

### 2.1.3. オブジェクトの生成

関数 IA\_3DV\_CreatePoint、IA\_3DV\_CreateQuad などにより表示したいオブジェクトを生成します。戻り値はオブジェクトIDです。  
現在、点、線、三角形、四角形、球が対応しています。  
位置と色またはマテリアル属性を指定します。

### 2.1.4. ビューの表示

ビューの表示、および更新を行います。生成したオブジェクトが反映されます。

### 2.1.5. ini ファイル入出力

視点、光源、座標線はダイアログ操作により設定しますが、ini ファイルへ入出力することもできます。  
これには IALIB の Eval 機能を使用していますので、Eval を有効にする必要があります。(2.1.1 節)

まず、ダイアログの設定値を ini ファイルの入出力対象とするために、関数 IA\_3DV\_RegisterParam を呼ぶ必要があります。  
そして関数 IA\_Eval\_LoadIniFile/ IA\_Eval\_SaveIniFile により ini ファイル入出力が行われます。



## 2.2. GUI

GUIでは以下を行えます。

- ・専用ダイアログ操作 ... 視点、光源、座標線を設定することができます。(2.2.1、2.2.2、2.2.3節)
- ・ビュー操作 ... 視点を設定することができます。(2.2.4節)

専用ダイアログはビューにあるツールバーのボタンを押下することで表示することができます。(図 2-2)

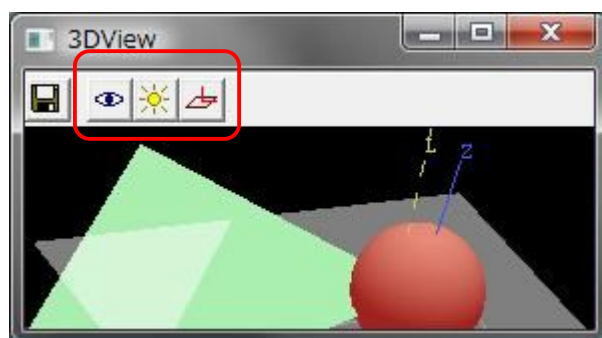


図 2-2 専用ダイアログを表示させるツールバーのボタン(赤枠)

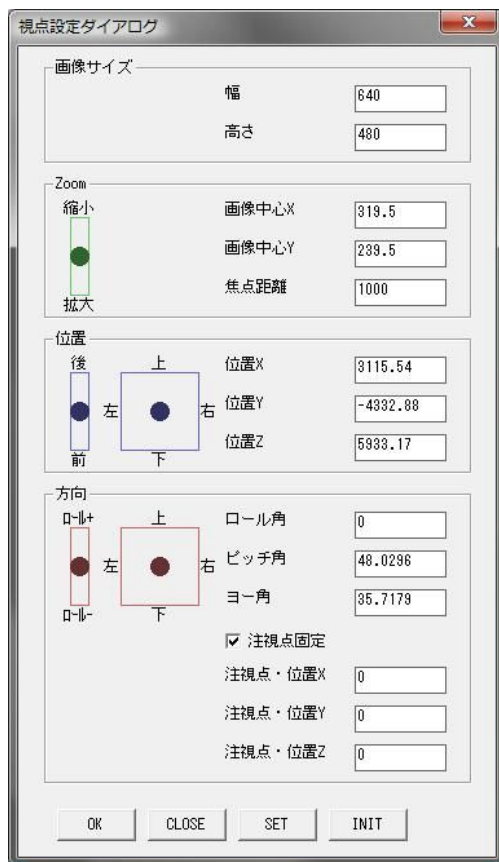
## 2.2.1. 視点設定ダイアログ

視点に関するパラメータを設定します。

右側のエディットボックスを編集して OK または SET ボタンを押下すると、ビューに反映されます。

左側の1次元/2次元トラックコントロールでも、パラメータをずらすことができます。

例えば Zoom の1次元トラックコントロールにて、中央にある●を下へドラッグすると、その移動量だけ画像が拡大表示されます。また対応するエディットボックスも更新されます。



「幅」、「高さ」、「画像中心 X/Y」はビューウィンドウのフレームをドラッグすることによっても、変更することができます。

左側の「上下左右前後」は、視点に対する相対的な方向です。トラックコントロールを「前」へずらすと、視点は前進します。

「注視点固定」を ON にすると、視点は「注視点・位置 X/Y/Z」を向くようになります。「ピッチ角」、「ヨー角」は設定することができなくなります。

図 2-3 視点設定ダイアログ

注視点固定の ON/OFF によって、トラックコントロールの操作によるパラメータの増減の仕方は異なります。

(1) 注視点固定 OFF

- ・位置の「上下左右」 … 「視点」が視線に垂直な平面上を移動。
- ・位置の「前後」 … 「視点」が視線上を移動。
- ・方向の「ロール±」 … 「視点」が視線を軸を中心に回転。
- ・方向の「上下左右」 … 「視線の先」が視点を中心とした球面上を移動。(「視点」の自転)

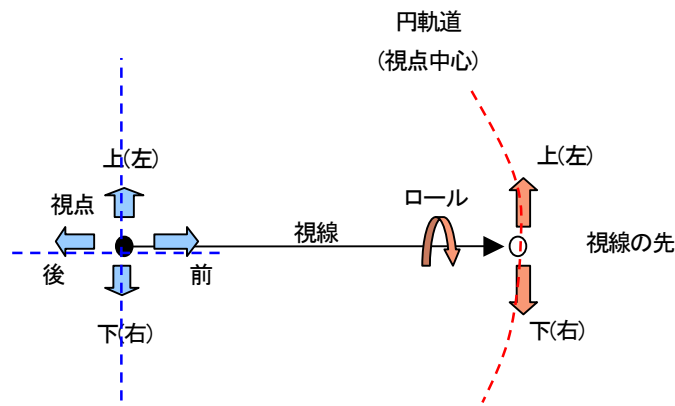


図 2-4 トラックコントロール操作によるパラメータの増減 (注視点無効)

(2) 注視点固定 ON

注視点に設定された座標は、常にビューに表示されます。

- ・位置の「上下左右」 … 「視点」が注視点を中心とした球面上を移動。(「視点」の公転)
- ・位置の「前後」 … 「視点」が視線上を移動。
- ・方向の「ロール±」 … 「視点」が視線を軸を中心に回転。
- ・方向の「上下左右」 … 注視点固定のため使用できません。

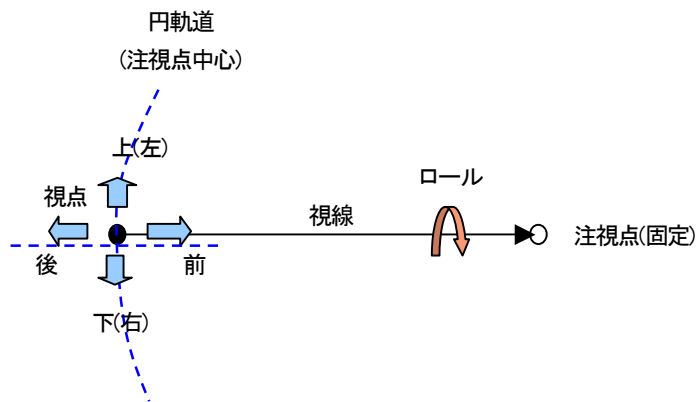


図 2-5 トラックコントロール操作によるパラメータの増減 (注視点有効)

## 2.2.2 光源設定ダイアログ

光源に関するパラメータを設定します。

エディットボックスを編集して OK または SET ボタンを押下すると、ビューに反映されます。  
トラックバーでも設定できます。

位置の「表示」を ON とすると、光源と原点を結んだ黄色い破線を表示することができます。(図 2-8)

位置は齊次座標で指定します。(X/W, Y/W, Z/W)が3次元座標となります。W=0 なら無限遠となります。

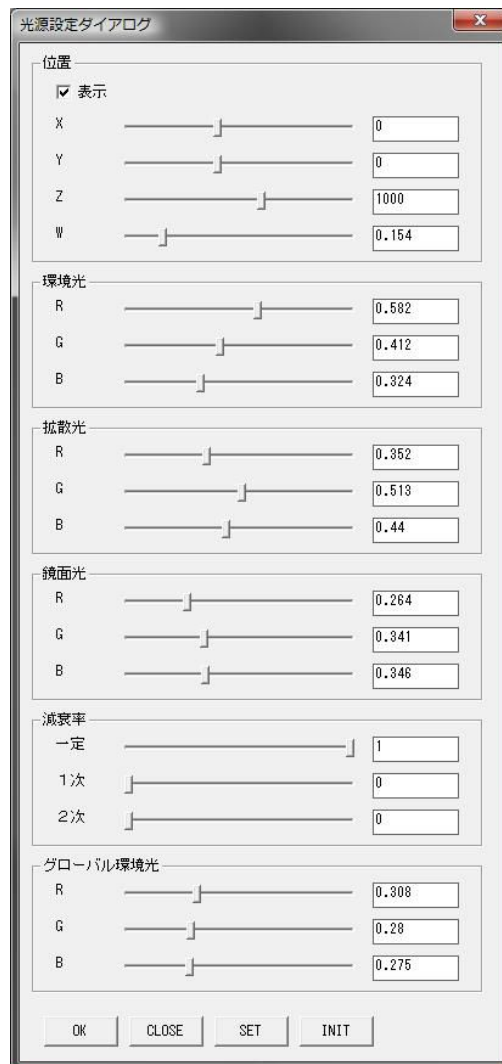


図 2-6 光源設定ダイアログ

### 2.2.3. 座標線設定ダイアログ

地面(XY 平面)および座標軸を表示して、オブジェクトの位置、方向、スケールなどを確認できます。(図 2-8)

エディットボックスを編集して OK または SET ボタンを押下すると、ビューに反映されます。

トラックバーでも設定できます。

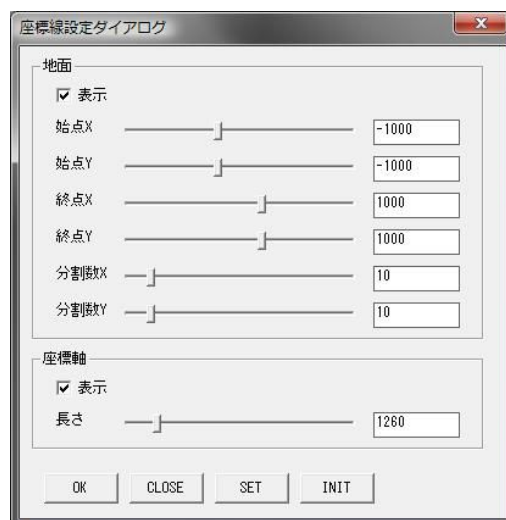


図 2-7 座標線設定ダイアログ

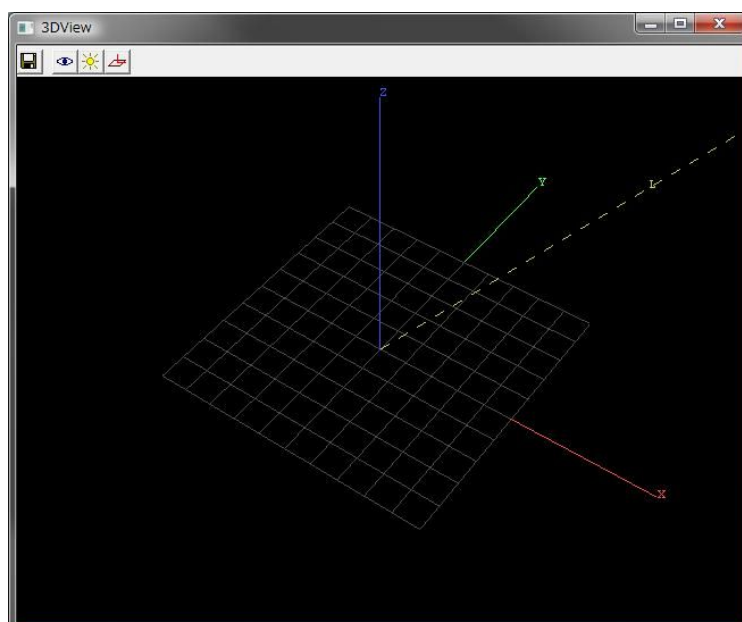


図 2-8 地面、座標軸、光源線

## 2.2.4. ビュー

ビューにマウス操作することにより、視点を変えることができます。

### 2.2.4.1. 視点の自転

座標軸の描画されていないところで、マウス左ボタンによるドラッグを行うと、視点が自転します。

自転は視点設定ダイアログでも行えます。(2.2.1 節)

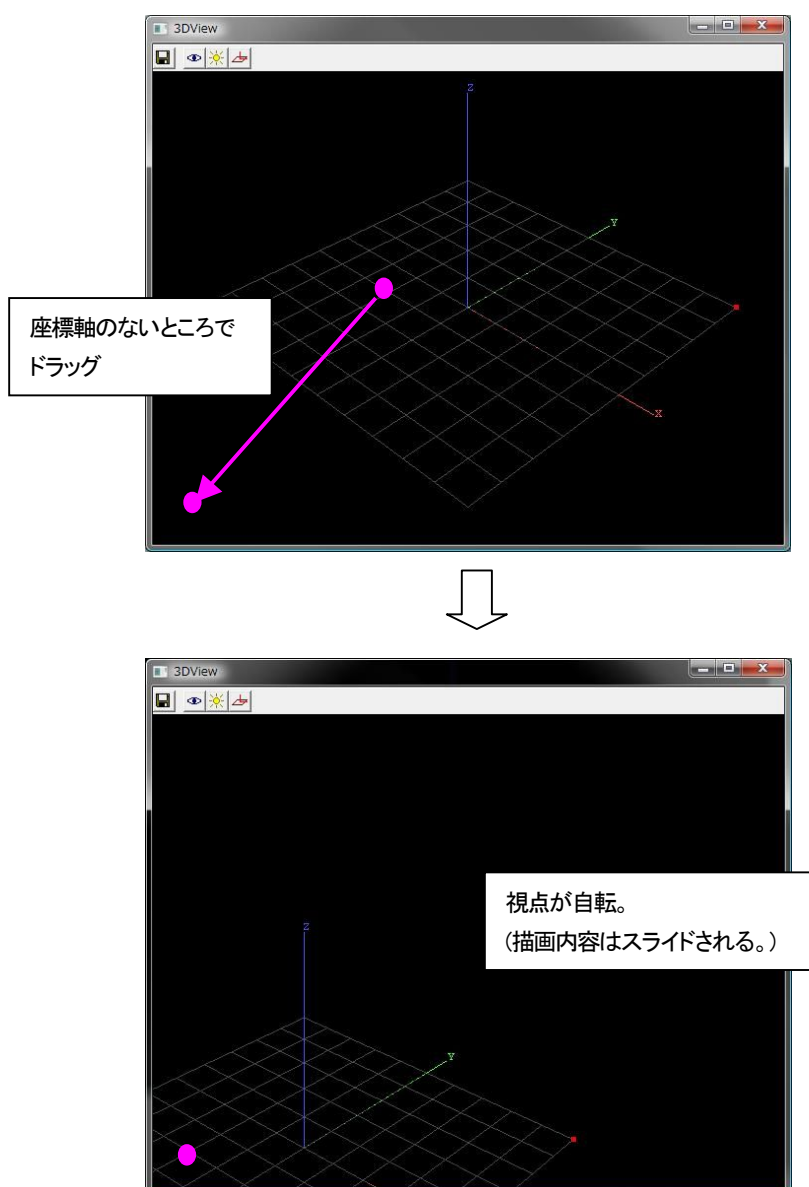


図 2-9 ドラッグ操作による視点の自転

### 2.2.4.2 視点の公転 (座標軸中心)

座標軸の描画されているところで、マウス左ボタンによる座標軸垂直方向のドラッグを行うと、視点がその座標軸を中心として公転します。

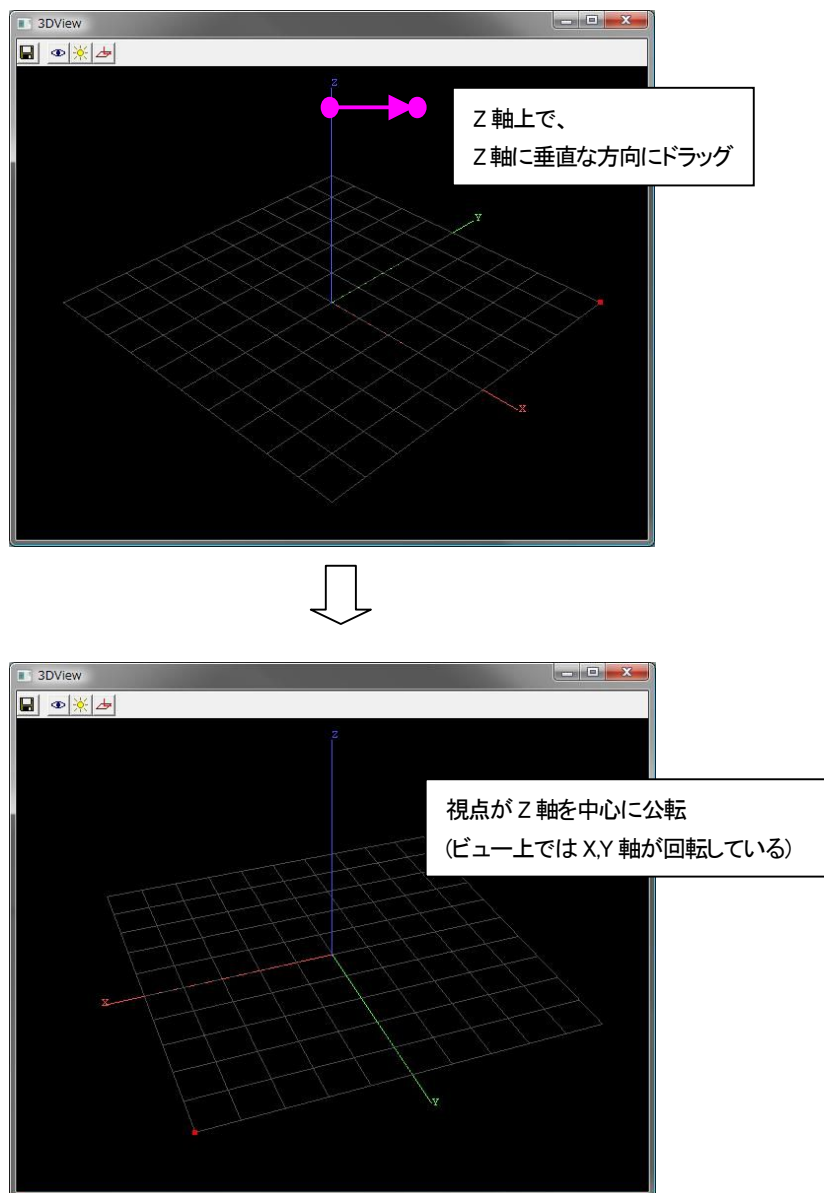


図 2-10 ドラッグ操作による視点の公転

### 2.2.4.3. 焦点距離の増減

マウスホイールを回転させると、焦点距離が増減します。  
 焦点距離が大きくなると、画角が狭くなり、画像が拡大されます。

焦点距離は視点設定ダイアログでも行えます。(2.2.1 節)

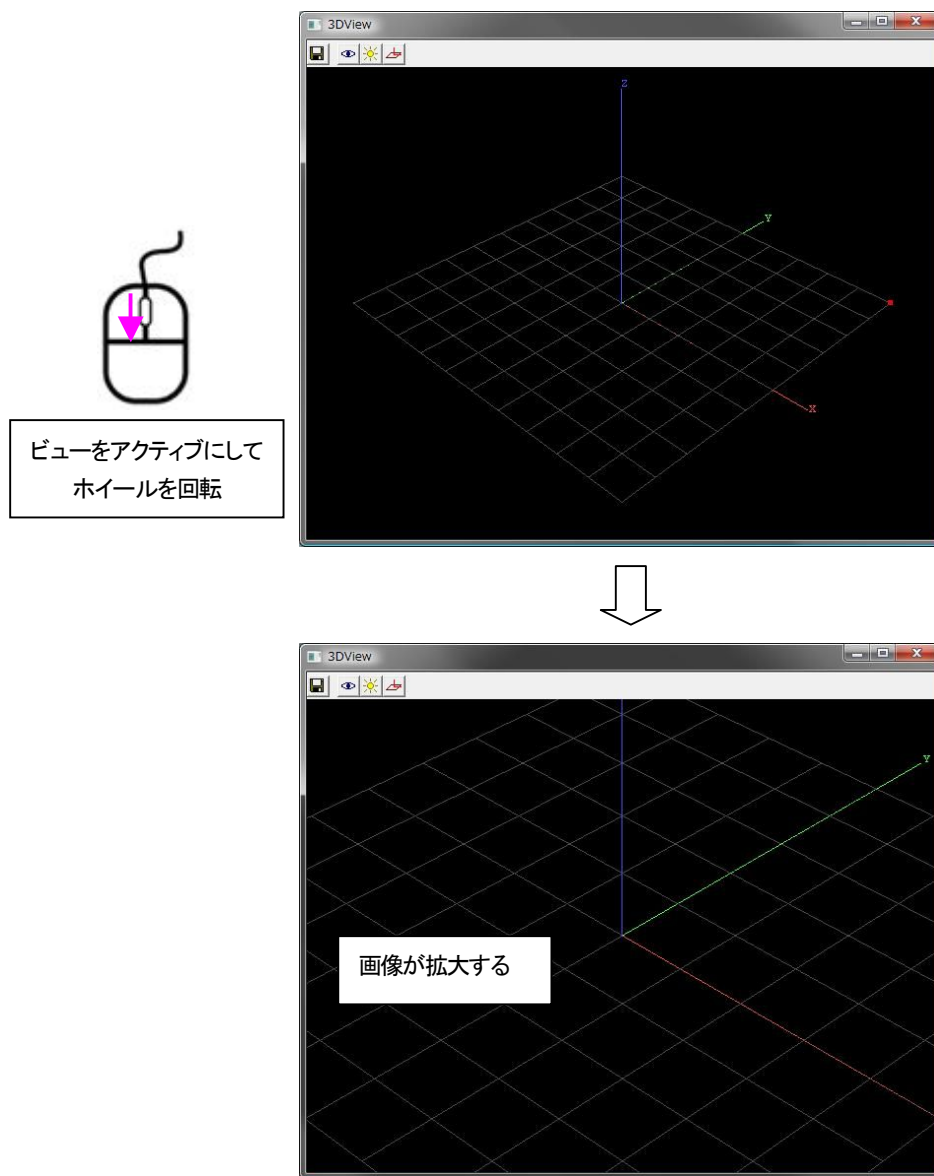


図 2-11 マウスホイール操作による視点の焦点距離増減



### 3. 関数

本章では関数インターフェースを記しています。  
 関連する列挙体、構造体は4章を参照してください。

区分	節番号	関数名	機能
有効/無効	3.1.1 節	IA_3DV_Open	有効化
	3.1.2 節	IA_3DV_Close	無効化
準備、表示	3.2.1 節	IA_3DV_Create	ビューの生成
	3.2.2 節	IA_3DV_Show	ビューの表示/非表示、表示更新
	3.2.3 節	IA_3DV_RegisterParam	保持情報の Eval 制御パラメータ登録
オブジェクト	3.3.1 節	IA_3DV_CreatePoint	オブジェクト(点)の生成
	3.3.2 節	IA_3DV_CreateLine	オブジェクト(線)の生成
	3.3.3 節	IA_3DV_CreateTriangle	オブジェクト(三角形)の生成
	3.3.4 節	IA_3DV_CreateQuad	オブジェクト(四角形)の生成
	3.3.5 節	IA_3DV_CreateSphere	オブジェクト(球)の生成
	3.3.6 節	IA_3DV_DeleteObject	オブジェクトの削除
	3.3.7 節	IA_3DV_DeleteAllObjects	オブジェクトの全削除
	3.3.8 節	IA_3DV_SetTexCoord	オブジェクトへのテクスチャ設定
	3.3.9 節	IA_3DV_ResetTexCoord	オブジェクトへのテクスチャ設定を無効化
テクスチャ	3.4.1 節	IA_3DV_CreateTexture	テクスチャの生成
	3.4.2 節	IA_3DV_DeleteTexture	テクスチャの削除
	3.4.3 節	IA_3DV_DeleteAllTextures	テクスチャの全削除
構造体初期化	3.5.1 節	IA_3DV_GetInitColor	IA_3DV_Color の初期化
	3.5.2 節	IA_3DV_GetInitMaterial	IA_3DV_Material の初期化
	3.5.3 節	IA_3DV_GetInitSphere	IA_3DV_GetInitSphere の初期化
	3.5.4 節	IA_3DV_GetInitTrOttoW	4x4 変換行列の初期化

表 3-1 3DView 関数一覧

### 3.1. 有効/無効

#### 3.1.1. 有効化 [IA\_3DV\_Open]

##### インタフェース

```
int IA_3DV_Open ();
```

##### リターン値

基本エラーコード

##### 機能

3DView を有効にします。

本関数の呼び出しは、IALIB を有効化した後(つまり関数 IA\_Open を呼んだ後)に行ってください。

### 3.1.2. 無効化 [IA\_3DV\_Close]

#### インタフェース

```
int IA_3DV_Close ();
```

#### リターン値

基本エラーコード

#### 機能

3DView を無効にします。

IALIB 内部にて確保したメモリの開放などの後処理を行いますので、有効にした場合には本関数を必ず呼び出してください。

本関数の呼び出しは、IALIB を無効化する前(つまり関数 IA\_Close を呼ぶ前)に行ってください。

## 3.2. 準備、表示

### 3.2.1. ビューの生成 [IA\_3DV\_Create]

#### インタフェース

```
int IA_3DV_Create ();
```

#### リターン値

0 以上 : ビューID  
0 未満 : 基本エラーコード (失敗)

#### 機能

ビューを生成します。

生成しただけでは表示されません。

オブジェクトを生成(3.3.1~3.3.5節)した後に関数 IA\_3DV\_Show(3.2.2節)を呼び出す必要があります。

ビューIDは複数のビューを識別するためのものです。

オブジェクトの生成などビュー毎に指示する場合にはこの ID を使用します。

ただし、現在はビューは1つしか生成できません。

### 3.2.2. ビューの表示/非表示、表示更新 [IA\_3DV\_Show]

#### インタフェース

```
int IA_3DV_Show (
    int viewID, int fShow
);
```

#### パラメータ

名称	タイプ	入/出	許容値	意味
viewID	int	入力		ビューID (3.2.1 節)
fShow	int	入力	0, 1	表示(1)/非表示(0)

#### リターン値

基本エラーコード

#### 機能

ビューを表示/非表示とし、表示の場合には表示内容を更新します。

通常は、オブジェクトを変更した場合に呼び出します。

### 3.2.3. 保持情報の Eval 制御パラメータ登録 [IA\_3DV\_RegisterParam]

#### インタフェース

```
int IA_3DV_RegisterParam (
    int viewID
);
```

#### パラメータ

名称	タイプ	入/出	許容値	意味
viewID	int	入力		ビューID (3.2.1 節)

#### リターン値

基本エラーコード

#### 機能

3DView が保持する以下情報

- ・視点
- ・光源
- ・座標線

を Eval の制御パラメータとして登録します。

IALIB の Eval 部の機能を使用しますので、事前に関数 IA\_Eval\_Enable により Eval を有効にする必要があります。

この登録では ini ファイル入出力機能のみの使用となります。

Eval の制御パラメータダイアログには表示されません。

ini ファイルへの入出力は他の制御パラメータと同様に

関数 IA\_Eval\_LoadIniFile 、 IA\_Eval\_SaveIniFile により行われます。

通常の制御パラメータ登録ではユーザーの用意したメモリを登録しますが、

この関数ではライブラリが内部にて動的に確保したメモリが登録されます。

この動的メモリはビューが破棄されたとき(現在は関数 IA\_3DV\_Close (3.1.2 節) が呼ばれたとき)に開放されます。

よって、ビューが破棄されてから ini ファイルの入出力は行わないでください。

Eval の詳細については、Eval マニュアルを参照してください。

### 3.3. オブジェクト設定

#### 3.3.1. オブジェクト(点)の生成 [IA\_3DV\_CreatePoint]

##### インタフェース

```
int IA_3DV_CreatePoint (
    int viewID, IA_3DV_Point *pPoint, float size, IA_3DV_Color *pColor, float trOtoW[16]
);
```

##### パラメータ

名称	タイプ	入/出	許容値	意味
viewID	int	入力		ビューID (3.2.1 節)
pPoint	IA_3DV_Point* (4.3 節)	入力		位置 (オブジェクト座標)
size	float	入力		表示サイズ
pColor	IA_3DV_Color* (4.1 節)	入力		色。 NULL ならデフォルトの色となります。
trOtoW	float[16]	入力		オブジェクト座標→世界座標の変換行列 (5.1、5.2 節)。 NULL なら無変換となります。

##### リターン値

0 以上 : オブジェクト ID  
0 未満 : 基本エラーコード (失敗)

##### 機能

点オブジェクトを生成します。

### 3.3.2. オブジェクト(線)の生成 [IA\_3DV\_CreateLine]

#### インタフェース

```
int IA_3DV_CreateLine (
    int viewID, IA_3DV_Point point[2], float size, IA_3DV_Color *pColor, float trOtoW[16]
);
```

#### パラメータ

名称	タイプ	入/出	許容値	意味
viewID	int	入力		ビューID (3.2.1節)
point	IA_3DV_Point[2] (4.3節)	入力		始点終点の位置 (オブジェクト座標)
size	float	入力		表示サイズ
pColor	IA_3DV_Color* (4.1節)	入力		色。 NULL ならデフォルトの色となります。
trOtoW	float[16]	入力		オブジェクト座標→世界座標の変換行列 (5.1、5.2節)。 NULL なら無変換となります。

#### リターン値

0 以上 : オブジェクト ID  
 0 未満 : 基本エラーコード (失敗)

#### 機能

線オブジェクトを生成します。



### 3.3.3. オブジェクト(三角形)の生成 [IA\_3DV\_CreateTriangle]

#### インタフェース

```
int IA_3DV_CreateTriangle (
    int viewID, IA_3DV_Point point[3], IA_3DV_Material *pMaterial, float trOtoW[16]
);
```

#### パラメータ

名称	タイプ	入/出	許容値	意味
viewID	int	入力		ビューID (3.2.1 節)
point	IA_3DV_Point[3] (4.3 節)	入力		頂点の位置 (オブジェクト座標)。 表面から見て左回りに指定してください。
pMaterial	IA_3DV_Material* (4.2 節)	入力		マテリアル特性。 NULL ならデフォルトのマテリアル特性となります。
trOtoW	float[16]	入力		オブジェクト座標→世界座標の変換行列 (5.1、5.2 節)。 NULL なら無変換となります。

#### リターン値

0 以上 : オブジェクト ID  
0 未満 : 基本エラーコード (失敗)

#### 機能

三角形オブジェクトを生成します。

ビューに表示される色は、マテリアル属性および光源の設定に依存します。

### 3.3.4. オブジェクト(四角形)の生成 [IA\_3DV\_CreateQuad]

#### インタフェース

```
int IA_3DV_CreateQuad (
    int viewID, IA_3DV_Point point[4], IA_3DV_Material *pMaterial, float trOtoW[16]
);
```

#### パラメータ

名称	タイプ	入/出	許容値	意味
viewID	int	入力		ビューID (3.2.1 節)
point	IA_3DV_Point[4] (4.3 節)	入力		頂点の位置 (オブジェクト座標) 表面から見て左回りに指定してください。
pMaterial	IA_3DV_Material* (4.2 節)	入力		マテリアル特性。 NULL ならデフォルトのマテリアル特性となります。
trOtoW	float[16]	入力		オブジェクト座標→世界座標の変換行列 (5.1、5.2 節)。 NULL なら無変換となります。

#### リターン値

0 以上 : オブジェクト ID  
0 未満 : 基本エラーコード (失敗)

#### 機能

四角形オブジェクトを生成します。

ビューに表示される色は、マテリアル属性および光源の設定に依存します。

### 3.3.5. オブジェクト(球)の生成 [IA\_3DV\_CreateSphere]

#### インタフェース

```
int IA_3DV_CreateSphere (
    int viewID, IA_3DV_Sphere *pSphere, IA_3DV_Material *pMaterial, float tr0toW[16]
);
```

#### パラメータ

名称	タイプ	入/出	許容値	意味
viewID	int	入力		ビューID (3.2.1節)
pSphere	IA_3DV_Sphere* (4.4節)	入力		位置、形状 (オブジェクト座標)
pMaterial	IA_3DV_Material* (4.2節)	入力		マテリアル特性。 NULL ならデフォルトのマテリアル特性となります。
tr0toW	float[16]	入力		オブジェクト座標→世界座標の変換行列 (5.1、5.2節)。 NULL なら無変換となります。

#### リターン値

0 以上 : オブジェクト ID  
0 未満 : 基本エラーコード (失敗)

#### 機能

球オブジェクトを生成します。

ビューに表示される色は、マテリアル属性および光源の設定に依存します。

### 3.3.6. オブジェクトの削除 [IA\_3DV\_DeleteObject]

#### インタフェース

```
int IA_3DV_DeleteObject (
    int viewID, int objID
);
```

#### パラメータ

名称	タイプ	入/出	許容値	意味
viewID	int	入力		ビューID (3.2.1 節)
objID	int	入力		オブジェクト ID (3.3.1~3.3.5 節)

#### リターン値

基本エラーコード

#### 機能

オブジェクトを削除します。

### 3.3.7. オブジェクトの全削除 [IA\_3DV\_DeleteAllObjects]

#### インタフェース

```
int IA_3DV_DeleteAllObjects (
    int viewID
);
```

#### パラメータ

名称	タイプ	入/出	許容値	意味
viewID	int	入力		ビューID (3.2.1 節)

#### リターン値

基本エラーコード

#### 機能

全てのオブジェクトを削除します。

### 3.3.8. オブジェクトへのテクスチャ設定 [IA\_3DV\_SetTexCoord]

#### インタフェース

```
int IA_3DV_SetTexCoord (
    int viewID, int objID, int texID, IA_PointF coord[]
);
```

#### パラメータ

名称	タイプ	入/出	許容値	意味
viewID	int	入力		ビューID (3.2.1節)
objID	int	入力		オブジェクトID (3.3.1~3.3.5節)
texID	int	入力		テクスチャID (3.4.1節)
coord	IA_PointF[]	入力		オブジェクト頂点のテクスチャ座標 (5.1.3節)

#### リターン値

基本エラーコード

#### 機能

オブジェクトにテクスチャを設定します。  
現在は三角形、四角形にのみ対応しています。

事前にオブジェクトおよびテクスチャを生成する必要があります。  
オブジェクトIDはオブジェクト生成関数の戻り値です。  
テクスチャIDはテクスチャ生成関数の戻り値です。

三角形の場合 coord[0]、coord[1]、coord[2]に頂点のテクスチャ座標を設定します。  
四角形の場合 coord[0]、coord[1]、coord[2]、coord[3]に頂点のテクスチャ座標を設定します。  
頂点の順番はオブジェクト生成時の順番と同じです。

### 3.3.9. オブジェクトへのテクスチャ設定を無効化 [IA\_3DV\_ResetTexCoord]

#### インタフェース

```
int IA_3DV_ResetTexCoord (
    int viewID, int objID
);
```

#### パラメータ

名称	タイプ	入/出	許容値	意味
viewID	int	入力		ビューID (3.2.1 節)
objID	int	入力		オブジェクト ID (3.3.1~3.3.5 節)

#### リターン値

基本エラーコード

#### 機能

オブジェクトへのテクスチャ設定(3.3.9 節)を無効化します。

## 3.4. テクスチャ設定

### 3.4.1. テクスチャの生成 [IA\_3DV\_CreateTexture]

#### インタフェース

```
int IA_3DV_CreateTexture (
    int viewID, int imgID
);
```

#### パラメータ

名称	タイプ	入/出	許容値	意味
viewID	int	入力		ビューID (3.2.1 節)
imgID	int	入力		テクスチャとしたい画像の画面 ID

#### リターン値

0 以上 : テクスチャ ID  
 0 未満 : 基本エラーコード (失敗)

#### 機能

テクスチャを生成します。

本関数では、3DView 内部にてテクスチャ用メモリが確保され、引数の画面 ID の画像がこのメモリへコピーされます。呼出し後、この画面を使用することはありませんので、開放しても問題ありません。テクスチャ機能は正常に動作します。

IALIB の処理ウィンドウは使用しません。画面全体がテクスチャとなります。

オブジェクトへテクスチャを設定するには、関数 IA\_3DV\_SetTexCoord (3.3.8 節) を呼び出します。



### 3.4.2. テクスチャの削除 [IA\_3DV\_DeleteTexture]

#### インタフェース

```
int IA_3DV_DeleteTexture (
    int viewID, int texID
);
```

#### パラメータ

名称	タイプ	入/出	許容値	意味
viewID	int	入力		ビューID (3.2.1 節)
texID	int	入力		テクスチャ ID (3.4.1 節)

#### リターン値

基本エラーコード

#### 機能

テクスチャを削除します。

### 3.4.3. テクスチャの全削除 [IA\_3DV\_DeleteAllTextures]

#### インタフェース

```
int IA_3DV_DeleteAllTextures (
    int viewID
);
```

#### パラメータ

名称	タイプ	入/出	許容値	意味
viewID	int	入力		ビューID (3.2.1 節)

#### リターン値

基本エラーコード

#### 機能

全てのテクスチャを削除します。

### 3.5. 構造体、配列の初期化

#### 3.5.1. IA\_3DV\_Color の初期化 [IA\_3DV\_GetInitColor]

##### インタフェース

```
int IA_3DV_GetInitColor (
    IA_3DV_Color *pColor
);
```

##### パラメータ

名称	タイプ	入/出	許容値	意味
pColor	IA_3DV_Color* (4.1節)	出力		色

##### リターン値

0

##### 機能

構造体 IA\_3DV\_Color の変数を初期化します。

### 3.5.2. IA\_3DV\_Material の初期化 [IA\_3DV\_GetInitMaterial]

#### インタフェース

```
int IA_3DV_GetInitMaterial (
    IA_3DV_Material * pMaterial
);
```

#### パラメータ

名称	タイプ	入/出	許容値	意味
pMaterial	IA_3DV_Material* (4.2節)	出力		マテリアル特性

#### リターン値

0

#### 機能

構造体 IA\_3DV\_Material の変数を初期化します。

### 3.5.3. IA\_3DV\_Sphere の初期化 [IA\_3DV\_GetInitSphere]

#### インタフェース

```
int IA_3DV_GetInitSphere (
    IA_3DV_Sphere* pObject
);
```

#### パラメータ

名称	タイプ	入/出	許容値	意味
pObject	IA_3DV_Sphere* (4.4節)	出力		球

#### リターン値

0

#### 機能

構造体 IA\_3DV\_Sphere の変数を初期化します。

### 3.5.4. 4x4 変換行列の初期化 [IA\_3DV\_GetInitTrOtoW]

#### インタフェース

```
int IA_3DV_GetInitTrOtoW (
    float trOtoW[16]
);
```

#### パラメータ

名称	タイプ	入/出	許容値	意味
trOtoW	float[16]	出力		4x4 変換行列

#### リターン値

0

#### 機能

無変換の 4x4 変換行列、つまり単位行列を返します。

trOtoW の書式については、5.2 節を参照してください。

## 4. 列挙体、構造体

### 4.1. 色

```
typedef struct {
    float r;
    float g;
    float b;
    float a;
} IA_3DV_Color;
```

色を表します。各要素 0~1 で指定します。

### 4.2. マテリアル属性

```
typedef struct {
    IA_3DV_Color ambient;           // 環境色 (R, G, B, A)
    IA_3DV_Color diffuse;          // 拡散色 (R, G, B, A)
    IA_3DV_Color specular;         // 鏡面色 (R, G, B, A)
    float          shininess;       // 鏡面指数
    IA_3DV_Color emission;         // 放射色 (R, G, B, A)
} IA_3DV_Material;
```

オブジェクトのマテリアル属性を表します。  
表示される色はこれと光源の設定値に依存します。

オブジェクトを半透明にするには、拡散色の A を 1 未満とします。

### 4.3. 3次元座標値

```
typedef struct {
    float x;
    float y;
    float z;
} IA_3DV_Point;
```

3次元座標値を表します。

### 4.4. 球

```
typedef struct {
    IA_3DV_Point center;    // 位置
    float radius;          // 半径
    int slices;             // 緯度方向分割数
    int stacks;             // 経度方向分割数
} IA_3DV_Sphere;
```

球の形状を表します。

分割数が大きいほど精度の高い球となります。



## 5. 補足

### 5.1. 座標

3DView では3つの座標を使用します。

#### 5.1.1. 世界座標

ビューごとに用意された3次元座標。(図 5-1)

この座標にて定義されたオブジェクト、視点、光源による各種計算を経て画像が作成されます。

#### 5.1.2. オブジェクト座標

オブジェクトごとに用意された3次元座標。(図 5-1)

視点、光源は世界座標にて定義されますが、オブジェクトはまずこのオブジェクト座標にて位置や大きさを指定します。

世界座標へ座標変換されて描画対象となります。

オブジェクト座標＝世界座標とすれば面倒な座標変換がなくなり簡略化できます。

#### 5.1.3. テクスチャ座標

テクスチャの位置を特定するための2次元座標。

画像座標と同様、テクスチャ座標も左上を原点、右方向を+X 方向、下方向を+Y 方向とします。

オブジェクトにテクスチャを設定する場合、

オブジェクトのオブジェクト座標を指定するだけでなく、このテクスチャ座標も設定する必要があります。

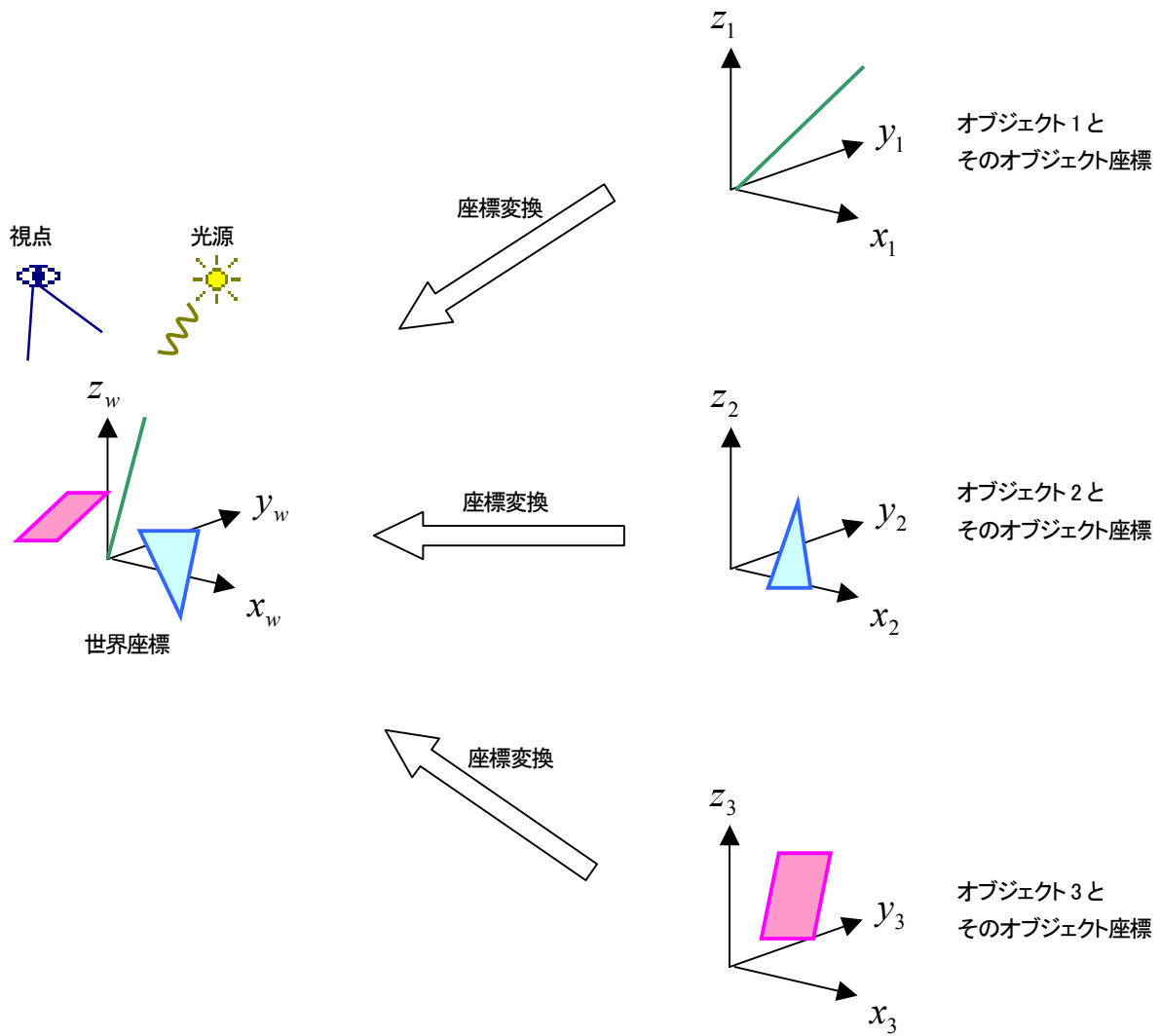


図 5-1 オブジェクト座標と世界座標

## 5.2. 座標変換

3DView では3次元座標を射影変換にて変換します。

射影変換は行列で表せます。

$$\begin{bmatrix} x_w \\ y_w \\ z_w \\ w_w \end{bmatrix} = \begin{bmatrix} m_{xx} & m_{xy} & m_{xz} & m_{xw} \\ m_{yx} & m_{yy} & m_{yz} & m_{yw} \\ m_{zx} & m_{zy} & m_{zz} & m_{zw} \\ m_{wx} & m_{wy} & m_{wz} & m_{ww} \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ z_o \\ w_o \end{bmatrix}$$

座標系 w の
変換行列
座標系 o の  
斉次座標

斉次座標

3DView ではこの変換行列を float 型の配列で表します。

$$\begin{bmatrix} x_w \\ y_w \\ z_w \\ w_w \end{bmatrix} = \begin{bmatrix} trOtoW[0] & trOtoW[1] & trOtoW[2] & trOtoW[3] \\ trOtoW[4] & trOtoW[5] & trOtoW[6] & trOtoW[7] \\ trOtoW[8] & trOtoW[9] & trOtoW[10] & trOtoW[11] \\ trOtoW[12] & trOtoW[13] & trOtoW[14] & trOtoW[15] \end{bmatrix} \begin{bmatrix} x_o \\ y_o \\ z_o \\ w_o \end{bmatrix}$$

float trOtoW[16];