

**REALAPS -Omni (Ver 1) API**  
**U-Omni-Glare.dll**  
**クラス ライブラリ説明書**

2023年6月17日

株式会社 ビジュアル・テクノロジー研究所 (略称: VTL)

I	U-Omni-Glare.dll について .....	1
1.1	U-Omni-Glare.dll の概要 .....	1
1.2	U-Omni-Glare.dll の呼び出し要件 .....	1
II	クラス ライブラリ説明 .....	2
1	Glare クラス .....	2
1.1	サブクラス AppendableData クラスとプロパティ .....	2
1.2	メソッド .....	2
1	getUGRValues() メソッド .....	3
2	getUGRValuesFromOXYZ() メソッド .....	5
3	getRealAppearance() メソッド .....	7
4	getRealAppearanceFromOXYZ() メソッド .....	8
5	savePictureRealAppearance() メソッド .....	8
6	savePictureGlare() メソッド .....	9
7	savePictureLightSource() メソッド .....	10
8	SaveRealapsJpeg() メソッド .....	10
2	UGR クラス .....	12
2.1	プロパティ .....	12
2.2	サブクラス Area クラスとプロパティ .....	12
2.3	サブクラス SightDirection クラスとプロパティ .....	13
2.4	使用例 .....	13
3	Glare.RealAppearance クラス .....	14
3.1	プロパティ .....	14
III	データ形式 .....	15
1	OXYZ ファイル .....	15
2	LM ファイル .....	15

# I U-Omni-Glare.dll について

## 1.1 U-Omni-Glare.dll の概要

U-Omni-Glare.dll は U-Omni-Glare の機能を、C#プログラムや 3D モデリングソフト Rhinoceros のプラグインツール Grasshopper から扱えるように、API 化したものです。

## 1.2 U-Omni-Glare.dll の呼び出し要件

1. 64bit の DLL のため、64bit 呼び出しを行ってください。  
フレームワークは .NET Framework 4.7.2 で作成されています。
2. DLL は REALAPS-Omni をインストールしたフォルダ下にある "RhinoPlugin" フォルダにインストールされます。  
Windows スタートメニューの "RhinoPlugin フォルダを開く" で、DLL のあるフォルダが開きます。

## II クラス ライブラリ説明

### 1 Glare クラス

#### 1.1 サブクラス AppendableData クラスとプロパティ

生成する REALAPS-Jpeg ファイルに色温度、緯度経度、その他自由な情報を追加するときに使います。

AppendableData のプロパティ一覧

	名前	型	説明
1	ColorTemperatureName	string	色温度(例 6500K)
2	CaptureDatetime	string	撮影日時(例: 2023/06/12 09:05:00)
3	Latitude	double	緯度(deg)
4	Longitude	double	経度(deg)
5	Notes	List<string>	

#### 1.2 メソッド

メソッド一覧

	名前	説明
1	getUGRValues()	輝度ファイルや輝度データから算出したグレア光源の直視 UGR 値とその立体角と UGR 値が戻ります。
2	getUGRValuesFromOXYZ()	OXYZ ファイルから算出したグレア光源の直視 UGR 値とその立体角と UGR 値が戻ります。
3	getRealAppearance()	輝度ファイルや輝度データとモニタ最大輝度と圧縮率から計算した sRGB 画像データを戻します。
4	getRealAppearanceFromOXYZ()	OXYZ ファイルの XYZ データとモニタ最大輝度と圧縮率から計算した sRGB 画像データを戻します。
5	savePictureRealAppearance()	sRGB 画像データからリアルアピランス画像を生成して PNG ファイルで保存します。
6	savePictureGlare()	UGRn 画像データから UGRn 分布を生成して PNG ファイルで保存します。
7	savePictureLightSource()	UGRn 画像データとグレア光源閾値から検出されたグレア光源画像を生成して PNG ファイルで保存します。
8	SaveRealapsJpeg	REALAPS-Jpeg ファイルを保存します。

## 1 getUGRValues() メソッド

輝度ファイルや輝度データから算出したグレア光源の直視 UGR 値とその立体角と UGR 値が戻ります。

1-1 輝度ファイルから求めた直視 UGR 画像データ(UGRn 値)と指定された範囲の直視 UGR 値とその立体角と UGR 値が戻ります。

UGR クラスに UGR 値を求める範囲と視線方向を指定することにより、その範囲のグレア光源の直視 UGR 値とその立体角と UGR 値が書き込まれて戻ります。

### 【構文】

```
enmResult getUGRValues(string LM_Filename, ref UGR data, out double[,] UGR_n)
```

### 【引数】

string LM\_Filename

- ・ 輝度画像データが CSV 形式で保存されている LM ファイル名

ref UGR data

- ・ 参照引数です。UGR 値を求める範囲と視線方向を指定した UGR クラスです。  
ここにグレア光源の直視 UGR 値と UGR 値が格納されて返ります。

out double[,] UGR\_n

- ・ 参照引数です。UGRn 値が格納されて返ります。

### 【戻り値】

返却値	意味
Success	成功しました。
Ranges_Are_Wrong	指定範囲が間違っています。
SightDirections_Are_Wrong	視線方向が間違っています。
UGR_SettingDataError	UGR データがありませんでした。
NotReadFileData	輝度データファイルが読み込めませんでした。
NotReadLmHeader	LM ファイルのヘッダが読み込めませんでした。
NotOmniData	全方位画像ではありません。
DataWidth_is_Zero	輝度データの幅サイズがゼロです。
DoErrorCalculatingNow	以前の計算が終了していません。
NullStatus	メソッドの状態異常が発生しました。
UnknownError	不明なエラー

1-2 輝度ファイルから求めた直視 UGR 画像データ(UGRn 値)と指定された範囲の直視 UGR 値とその立体角と UGR 値が戻ります。

UGR クラスに UGR 値を求める範囲と視線方向をまとめて配列で指定することにより、その範囲のグレア光源の直視 UGR 値とその立体角と UGR 値が書き込まれて戻ります。

**【構文】**

```
enmResult getUGRValues(string LM_Filename, ref UGR[ ] data, out double[,] UGR_n)
```

**【引数】**

string LM\_Filename

- ・ 輝度画像データが CSV 形式で保存されている LM ファイル名

ref UGR[ ] data

- ・ 参照引数です。UGR 値を求める範囲と視線方向を指定した UGR クラスの配列です。ここにグレア光源の直視 UGR 値と UGR 値が格納されて返ります。

out double[,] UGR\_n

- ・ 参照引数です。UGR 値を求める範囲と視線方向を指定した UGR クラスの配列です。

**【戻り値】**

返却値	意味
Success	成功しました。
Ranges_Are_Wrong	指定範囲が間違っています。
SightDirections_Are_Wrong	視線方向が間違っています。
UGR_SettingDataError	UGR データがありませんでした。
NotReadFileData	輝度データファイルが読み込めませんでした。
NotReadLmHeader	LM ファイルのヘッダが読み込めませんでした。
NotOmniData	全方位画像ではありません。
DataWidth_is_Zero	輝度データの幅サイズがゼロです。
DoErrorCalculatingNow	以前の計算が終了していません。
NullStatus	メソッドの状態異常が発生しました。
UnknownError	不明なエラー

## 2 getUGRValuesFromOXYZ() メソッド

OXYZ ファイルから算出したグレア光源の直視 UGR 値とその立体角と UGR 値が戻ります。

2-1 OXYZ ファイルの輝度データから求めた直視 UGR 画像データ(UGRn 値)と指定された範囲の直視 UGR 値とその立体角と UGR 値が戻ります。

UGR クラスに UGR 値を求める範囲と視線方向を指定することにより、その範囲のグレア光源の直視 UGR 値とその立体角と UGR 値が書き込まれて戻ります。

### 【構文】

```
enmResult getUGRValuesFromOXYZ(string oxyz_Filename, ref UGR data, out double[,] UGR_n)
```

### 【引数】

string oxyz\_Filename

- ・ OXYZ ファイル名です。

ref UGR data

- ・ 参照引数です。UGR 値を求める範囲と視線方向を指定した UGR クラスです。  
ここにグレア光源の直視 UGR 値と UGR 値が格納されて返ります。

out double[,] UGR\_n

- ・ 参照引数です。UGRn 値が格納されて返ります。

### 【戻り値】

返却値	意味
Success	成功しました。
Ranges_Are_Wrong	指定範囲が間違っています。
SightDirections_Are_Wrong	視線方向が間違っています。
UGR_SettingDataError	UGR データがありませんでした。
NotReadFileData	輝度データファイルが読み込めませんでした。
NotReadOxyzHeader	OXYZ ファイルからのヘッダが読み込めませんでした
NotOmniData	全方位画像ではありません。
DataWidth_is_Zero	輝度データの幅サイズがゼロです。
DoErrorCalculatingNow	以前の計算が終了していません。
NullStatus	メソッドの状態異常が発生しました。
UnknownError	不明なエラー

2-2 OXYZ ファイルの輝度データから求めた直視 UGR 画像データ(UGRn 値)と指定された範囲の直視 UGR 値とその立体角と UGR 値が戻ります。

UGR クラスに UGR 値を求める範囲と視線方向をまとめて配列で指定することにより、その範囲のグレア光源の直視 UGR 値とその立体角と UGR 値が書き込まれて戻ります。

**【構文】**

```
enmResult getUGRValuesFromOXYZ(string oxyz_Filename, ref UGR[ ] data, out double[,] UGR_n)
```

**【引数】**

string oxyz\_Filename

- ・ OXYZ ファイル名です。

ref UGR[ ] data

- ・ 参照引数です。UGR 値を求める範囲と視線方向を指定した UGR クラスの配列です。ここにグレア光源の直視 UGR 値と UGR 値が格納されて返ります。

out double[,] UGR\_n

- ・ 参照引数です。UGRn 値が格納されて返ります。

**【戻り値】**

返却値	意味
Success	成功しました。
Ranges_Are_Wrong	指定範囲が間違っています。
SightDirections_Are_Wrong	視線方向が間違っています。
UGR_SettingDataError	UGR データがありませんでした。
NotReadFileData	輝度データファイルが読み込めませんでした。
NotReadOxyzHeader	OXYZ ファイルからのヘッダが読み込めませんでした
NotOmniData	全方位画像ではありません。
DataWidth_is_Zero	輝度データの幅サイズがゼロです。
DoErrorCalculatingNow	以前の計算が終了していません。
NullStatus	メソッドの状態異常が発生しました。
UnknownError	不明なエラー



### 3 getRealAppearance() メソッド

輝度ファイルとモニタ最大輝度と圧縮率から計算した sRGB 画像データを戻します。sRGB とはガンマ 2.2 補正を掛けた 0~255 の RGB の値です。

モニタ最大輝度と圧縮率は、Glare.RealAppearance クラスの変数を用いて設定します。

#### 【構文】

```
enmResult getRealAppearance(string LM_Filename, out byte[ ][,] sRGB)
```

#### 【引数】

string LM\_Filename

- ・ LM ファイル名です

out byte[ ][,] sRGB

- ・ 参照引数です。sRGB 画像データが格納されます。

#### 【戻り値】

返却値	意味
Success	成功しました。
NotReadFileData	データが読み込めませんでした
NotReadLmHeader	LM データのヘッダが読み込めませんでした
NotOmniData	全方位画像ではありません。
DoErrorCalculatingNow	以前の計算が終了していません。
NullStatus	メソッドの状態異常が発生しました。
UnknownError	不明なエラー

#### 4 getRealAppearanceFromOXYZ() メソッド

OXYZ ファイルの XYZ データとモニタ最大輝度と圧縮率から計算した sRGB 画像データを戻します。sRGB とはガンマ 2.2 補正を掛けた 0~255 の RGB の値です。

モニタ最大輝度と圧縮率は、Glare.RealAppearance クラスの変数を用いて設定します。

##### 【構文】

```
enmResult getRealAppearanceFromOXYZ(string oxyz_filename, out byte[ ][,] sRGB)
```

##### 【引数】

string oxyz\_filename

- ・ OXYZ ファイル名

out byte[ ][,] sRGB

- ・ 参照引数です。sRGB 画像データが格納されます。

##### 【戻り値】

返却値	意味
Success	成功しました。
NotReadFileData	XYZ データが読み込めませんでした
NotReadOxyzHeader	OXYZ ファイルのヘッダが読み込めませんでした
NotOmniData	全方位画像ではありません。
DoErrorCalculatingNow	以前の計算が終了していません。
NullStatus	メソッドの状態異常が発生しました。
UnknownError	不明なエラー

#### 5 savePictureRealAppearance() メソッド

sRGB 画像データからリアルアピランス画像を生成して PNG ファイルで保存します。

##### 【構文】

```
enmResult savePictureRealAppearance(string savePNGfilename, byte[ ][,] sRGB)
```

##### 【引数】

string savePNGfilename

- ・ 保存する PNG ファイル名

byte[ ][,] sRGB

- ・ sRGB 画像データ配列

【戻り値】

返却値	意味
Success	成功しました。
ErrorNonData	データがありません
Ranges_Are_Wrong	スケールの最大値と最小値が合っていません

## 6 savePictureGlare() メソッド

直視 UGR 画像データ(UGRn)と sRGB 画像データから UGRn 分布画像を生成して PNG ファイルで保存します。

【構文】

```
enmResult savePictureGlare(string savePNGfilename, double[,] UGR_n, byte[ ][,] sRGB,
bool WithColorBar, double min, double max)
```

【引数】

string savePNGfilename

- ・ 保存する PNG ファイル名

double[,] UGR\_n

- ・ 直視 UGR 画像データ(UGRn)配列

byte[ ][,] sRGB

- ・ sRGB 画像データ配列

bool WithColorBar

- ・ 保存画像にスケールバーを付ける場合は true、付けない時は false

double min

- ・ 画像スケールの最小値

double max

- ・ 画像スケールの最大値

【戻り値】

返却値	意味
Success	成功しました。
ErrorNonData	データがありません
Ranges_Are_Wrong	スケールの最大値と最小値が合っていません

## 7 savePictureLightSource() メソッド

直視 UGR 画像データ(UGRn)と UGRn 閾値と sRGB 画像データから検出されたグレア光源画像を生成して PNG ファイルで保存します。

### 【構文】

```
enmResult savePictureLightSource(string savePNGfilename, double[,] UGR_n, byte[ ][,] sRGB,
double threshold_UGR_N)
```

### 【引数】

string savePNGfilename

- ・ 保存する PNG ファイル名

double[,] UGR\_n

- ・ 直視 UGR 画像データ(UGRn)配列

byte[ ][,] sRGB

- ・ sRGB 画像データ配列

double threshold\_UGR\_N

- ・ グレア光源としての UGRn 閾値

### 【戻り値】

返却値	意味
Success	成功しました。
ErrorNonData	データがありません。
Ranges_Are_Wrong	スケールの最大値と最小値が合っていません。

## 8 SaveRealapsJpeg() メソッド

REALAPS-Jpeg ファイルを保存します。

### 【構文】

```
enmResult SaveRealapsJpeg(
string oxyzFilename, string realapsJpegFilename, int width, AppendableData appendableData = null,
double monitorMaxLuminance = double.NaN, double compressionRatio = double.NaN, bool isRaOnly = false)
```

【引数】

string oxyzFilename

- REALAS-Jpeg 作成のために読み込む OXYZ ファイルパス(ファイル名)です。

string realapsJpegFilename

- 保存する REALAPS-Jpeg のファイルパス(ファイル名)

width

- 保存する REALAPS-Jpeg の画像の幅(pixel)

AppendableData appendableData

- 生成する REALAPS-Jpeg ファイルに追加するデータを格納した AppendableData オブジェクト
- 格納できるデータは、色温度、撮影日時、緯度、経度、その他などです。
- 省略可能です。省略時は、新規に REALAPS-Jpeg を作成するときはこれらのデータは保存されず、既存の REALAPS-Jpeg に上書きする場合は、既存のデータが引き継がれます。

monitorMaxLuminance

- リアルアピランス画像を作成するときに用いる画面最大輝度の値です。
- 省略可能です。省略時は OXYZ ファイルに格納されている値を用います。

compressionRatio

- リアルアピランス画像を作成するときに用いる圧縮率の値です。
- 省略可能です。省略時は OXYZ ファイルに格納されている値を用います。

isRaOnly

- true REALAPS-Jpeg に格納されている各種データは変えずに、リアル・アピランス画像のみを差し替えます。
- false 通常の REALAPS-Jpeg を生成します。
- 省略可能です。省略時は false となります。

【戻り値】

返却値	意味
Success	成功しました。
FileNotExist	OXYZ ファイルが見つかりません。
NotReadFileData	OXYZ ファイルが読み込めません。
NotReadOxyzHeader	OXYZ ファイルのヘッダが読み込めません。
NotOmniData	全方位画像ではありません。
DoErrorCalculatingNow	計算時にエラーが発生しました。
NonXYZ_Data	XYZ データがありません。
UnknownError	不明なエラー

## 2 UGR クラス

グレア光源の直視 UGR 値とその立体角と UGR 値を求めるために利用します。

UGR 値を求める範囲とその範囲より求めたグレア光源の直視 UGR 値、その立体角、UGR 値が保持されます。

### 2.1 プロパティ

プロパティ一覧

	名前	型	説明
1	UGR_LookingDirectly	Double	求めた枠内検出グレア光源の直視 UGR 値を保持しています。
2	Steradian_LookingDirectly	Double	求めた枠内検出グレア光源の直視 UGR の立体角を保持しています。
3	UGR_SightLine	Double	求めた枠内検出グレア光源の UGR 値を保持しています。
4	Threshold_UGR_N	Double	直視 UGR 値、立体角、UGR 値を求めるために設定されたグレア光源の閾値を保持します。
5	area	Area	UGR クラス内に定義された Area クラスによって定義された UGR 値などを求める範囲を保持します。
6	sightDirection	SightDirection	UGR クラス内に定義された SightDirection クラスによって定義された視線方向を保持します。

### 2.2 サブクラス Area クラスとプロパティ

UGR クラス内に定義された Area クラスによって定義された UGR 値などを求める範囲が保持されるクラスです。

プロパティ一覧

	名前	型	説明
1	rectangle	Rectangle	UGR 値などを求める矩形範囲を設定します。
2	AllArea	Bool	TRUE の場合は画像の全エリアが計算対象となります。

## 2.3 サブクラス SightDirection クラスとプロパティ

UGR クラス内に定義された SightDirection クラスによって定義された視線方向と画像幅が保持されるクラスです。

プロパティ一覧

	名前	型	説明
1	Pixel	Point	視線を向ける対象の画像ピクセル座標を設定します。
2	Azimuth	Double	視線を向ける方向の方位角を設定します。 $-180 \leq \text{Azimuth} < 180$ の範囲です。
3	Elevation	Double	視線を向ける方向の仰角を設定します。 $-90 \leq \text{Elevation} \leq 90$ の範囲です。上向きが正です。
4	Width	Int	視線を向ける画像の幅を設定します。

## 2.4 使用例

```

Glare glare = new Glare ();
UGR ugr = new UGR (); //画像幅をセットしない場合は、方位角と仰角を設定すればいい。
ugr.sightDirection.Azimuth = 0.0;
ugr.sightDirection.Elevation = 0.0;
enmResult glare.getUGRValues(@"C:\temp¥Sample.lm", ref ugr, out double[,] ugr_n);
string text = result.ToString () + "¥r¥n";
text += "検出枠範囲の左上の点 (" + ugr.area.rectangle.X.ToString () + ", " +
ugr.area.rectangle.Y.ToString () + ")¥r¥n";
text += "検出枠範囲の幅: " + ugr.area.rectangle.Width.ToString () + " 高さ: " +
ugr.area.rectangle.Height.ToString () + "¥r¥n";
text += "視線方向 (" + ugr.sightDirection.Pixel.X.ToString () + ", " +
ugr.sightDirection.Pixel.Y.ToString () + ")¥r¥n";
text += "視線方向 方位角:" + ugr.sightDirection.Azimuth.ToString () + " 仰角: " +
ugr.sightDirection.Elevation.ToString () + "¥r¥n";
text += "枠内検出グレア光源の直視 UGR=" + ugr.UGR_LookingDirectly.ToString () + "¥r¥n";
text += "枠内検出グレア光源の直視 UGR の立体角=" + ugr.Steradian_LookingDirectly.ToString () +
"¥r¥n";
text += "枠内検出グレア光源の UGR=" + ugr.UGR_SightLine.ToString () + "¥r¥n";
Print(text);

```

### 3 Glare.RealAppearance クラス

リアルアピランス画像を作成するときの画面最大輝度値と圧縮率を設定するクラスです。  
画面最大輝度と圧縮率を自動計算させるフラグも持っています。

#### 3.1 プロパティ

プロパティ一覧

	名前	型	説明
1	IsAuto	Bool	パラメータを自動計算するかどうかのフラグです。
2	MonitorMaxLuminance	Double	画面最大輝度値を設定します。自動計算を選択した場合は自動計算結果が設定されます。
3	CompressionRatio	Double	圧縮率を設定します。自動計算を選択した場合は自動計算結果が設定されます。



### III データ形式

#### 1 OXYZ ファイル

輝度+XYZ 表色系の色度分布が記述された測光色画像データで、REALAPS 関連ソフトで生成されるデータ形式です。

#### 2 LM ファイル

輝度分布が記述された輝度画像データで、CSV データ形式です。

先頭の一行目に、画像データの横幅と高さおよび画角が CSV 形式で書かれています。

二行目以降は、輝度データが高さ順に上から CSV 形式で並んでいます。

