

**REALAPS -Omni (Ver 1) API**  
**REALAPS -Omni.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	REALAPS-Omni.dll について .....	1
1.1	REALAPS-Omni.dll の概要 .....	1
1.2	REALAPS-Omni.dll の呼び出し要件 .....	1
II	クラス ライブラリ説明 .....	2
1	Omni クラス .....	2
1.1	サブクラス AppendableData クラス .....	2
1.2	メソッド .....	2
1	getEstimationValues() メソッド .....	3
2	getEstimationValuesFromOXYZ() メソッド .....	6
3	getIlluminance() メソッド .....	8
4	getIlluminanceFromOXYZ() メソッド .....	9
5	getRealAppearance() メソッド .....	10
6	getRealAppearanceFromOXYZ() メソッド .....	11
7	savePictureLuminance() メソッド .....	12
8	savePictureNB() メソッド .....	13
9	savePictureRealAppearance() メソッド .....	13
10	savePictureIlluminance() メソッド .....	14
11	getOxyzFromJpegs() メソッド .....	15
12	SaveRealapsJpeg() メソッド .....	15
2	EstimatingBR クラス .....	17
2.1	プロパティ .....	17
2.2	サブクラス Area クラスとプロパティ .....	17
2.3	使用例 .....	18
3	JpegParam クラス .....	19
3.1	プロパティ .....	19
3.2	使用例 .....	19
III	データ形式 .....	20
1	OXYZ ファイル .....	20
2	LM ファイル .....	20

# I REALAPS-Omni.dll について

## 1.1 REALAPS-Omni.dll の概要

REALAPS-Omni.dll は REALAPS-Omni 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 REALAPS-Omni.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 Omni クラス

#### 1.1 サブクラス AppendableData クラス

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	ColorTemperqttureName	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	getEstimationValues()	輝度ファイルや輝度データから求めた明るさ画像データ(NB 値)と、指定された範囲の明るさ感推定値が戻ります。
2	getEstimationValuesFromOXYZ()	OXYZ ファイルから求めた明るさ画像データ(NB 値)と、指定された範囲の明るさ感推定値が戻ります。
3	getIlluminance()	輝度ファイルや輝度データと反射率から求めた照度画像データ(lx)が戻ります。
4	getIlluminanceFromOXYZ()	OXYZ ファイルの輝度データと反射率から求めた照度画像データ(lx)が戻ります。
5	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。
6	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。
7	savePictureLuminance()	輝度画像データから輝度画像を生成して PNG ファイルで保存します。
8	savePictureNB()	明るさ画像データ(NB)から明るさ画像と明るさ検討画像を生成して PNG ファイルで保存します。

9	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。
10	savePictureIlluminance()	照度画像データから照度画像を生成して PNG ファイルで保存します。
11	getOxyzFromJpegs()	1 枚以上の Jpeg 画像から XYZ データを求めて OXYZ ファイルを保存します。
12	SaveRealapsJpeg	REALAPS-Jpeg ファイルを保存します。

## 1 getEstimationValues() メソッド

輝度ファイルや輝度データから求めた明るさ画像データ(NB 値)と指定された範囲の明るさ感推定値が戻ります。

1-1 輝度ファイルから求めた明るさ画像データ(NB 値)と指定された範囲の明るさ感推定値が戻ります。

EsrtimatingBR クラスに明るさ感推定値を求める範囲を指定することにより、その範囲の明るさ感推定値と各 NB 値範囲の割合が書き込まれて戻ります。

### 【構文】

```
enmResult getEstimationValues(
string LM_Filename, ref double[,] NB, ref EstimatingBR data, AreaType areaType = AreaType.Office)
```

### 【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
- ・ LM ファイルのデータ形式については III データ形式 (20 ページ) を参照してください。

ref double[,] NB

- ・ 参照引数です。明るさ NB 値が格納されて返ります。

ref EstimatingBR data

- ・ 参照引数です。明るさ感推定値の測定範囲を指定した EsrtimatingBR クラスです。
- ・ ここに明るさ感推定値と各明るさ検討用データの割合が格納されて返ります。

AreaType areaType

- ・ enum の AreaType を指定します。  
Office, Living, Museum のいずれかを設定できます。
- ・ 省略可能です。省略時は Office が設定されます。

【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。
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。
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明るさ画像データ(NB 値)と指定された範囲の明るさ感推定値が戻ります。

EsrtimatingBR クラスに明るさ感推定値を求める範囲をまとめて配列で指定することにより、その範囲の明るさ感推定値と各 NB 値範囲の割合が書き込まれて戻ります。

【構文】

```
enmResult getEstimationValues(
string LM_Filename, ref double[,] NB, ref EstimatingBR[] data, AreaType areaType = AreaType.Office)
```

【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  
LM ファイルのデータ形式については III データ形式 (20 ページ) を参照してください。

ref double[,] NB

- 参照引数です。明るさ NB 値が格納されて返ります。

ref EstimatingBR[] data

- 参照引数です。明るさ感推定値の測定範囲を指定した EsrtimatingBR クラスの配列です。
- ここに明るさ感推定値と各明るさ検討用データの割合が格納されて返ります。

AreaType areaType

- enum の AreaType を指定します。  
Office, Living, Museum のいずれかを設定できます。
- 省略可能です。省略時は Office が設定されます。

【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。
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。

NullStatus	メソッドの状態異常が発生しました。
UnknownError	不明なエラー

- 1-3 輝度画像データから求めた明るさ画像データ(NB 値)と指定された範囲の明るさ感推定値が戻ります。  
**EsrtimatingBR** クラスに明るさ感推定値を求める範囲を指定することにより、その範囲の明るさ感推定値と各 NB 値範囲の割合が書き込まれて戻ります。

**【構文】**

```
enmResult getEstimationValues(  
float[,] luminance, ref double[,] NB, ref EstimatingBR[] data, AreaType areaType = AreaType.Office)
```

**【引数】**

float[,] luminance

- ・ 輝度画像データ配列です。

ref double[,] NB

- ・ 参照引数です。明るさ NB 値が格納されて返ります。

ref EstimatingBR data

- ・ 参照引数です。明るさ感推定値の測定範囲を指定した **EsrtimatingBR** クラスです。ここに明るさ感推定値と各明るさ検討用データの割合が格納されて返ります。

AreaType areaType

- ・ enum の AreaType を指定します。  
Office, Living, Museum のいずれかを設定できます。
- ・ 省略可能です。省略時は Office が設定されます。

**【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。
NoLuminanceData	輝度データが読み込めませんでした。
NullStatus	メソッドの状態異常が発生しました。
UnknownError	不明なエラー

- 1-4 輝度画像データから求めた明るさ画像データ(NB 値)と指定された範囲の明るさ感推定値が戻ります。  
**EsrtimatingBR** クラスに明るさ感推定値を求める範囲をまとめて配列で指定することにより、その範囲の明るさ感推定値と各 NB 値範囲の割合が書き込まれて戻ります。

【構文】

```
enmResult getEstimationValues(  
float[,] luminance, ref double[,] NB, ref EstimatingBR[] data, AreaType areaType = AreaType.Office)
```

【引数】

float[,] luminance

- ・ 輝度画像データ配列です。

ref double[,] NB

- ・ 参照引数です。明るさ NB 値が格納されて返ります。

ref EstimatingBR[] data

- ・ 参照引数です。明るさ感推定値の測定範囲を指定した **EsrtimatingBR** クラスの配列です。

ここに明るさ感推定値と各明るさ検討用データの割合が格納されて返ります。

AreaType areaType

- ・ enum の AreaType を指定します。  
Office, Living, Museum のいずれかを設定できます。
- ・ 省略可能です。省略時は Office が設定されます。

【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。
NoLuminanceData	輝度データが読み込めませんでした。
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 getEstimationValuesFromOXYZ() メソッド

OXYZ ファイルから求めた明るさ画像データ(NB 値)と指定された範囲の明るさ感推定値が返ります。

2-1 OXYZ ファイルの輝度データから求めた明るさ画像データ(NB 値)と指定された範囲の明るさ感推定値が返ります。

EsrtimatingBR クラスに明るさ感推定値を求める範囲を指定することにより、その範囲の明るさ感推定値と各 NB 値範囲の割合が書き込まれて返ります。

【構文】

```
enmResult getEstimationValuesFromOXYZ(  
string oxyz_filename, ref double[,] NB, ref EstimatingBR data, AreaType areaType = AreaType.Office)
```

【引数】

string oxyz\_Filename

- ・ OXYZ ファイル名です。

ref double[,] NB

- ・ 参照引数です。明るさ NB 値が格納されて返ります。

ref EstimatingBR data

- ・ 参照引数です。明るさ感推定値の測定範囲を指定した **EsrtimatingBR** クラスです。ここに明るさ感推定値と各明るさ検討用データの割合が格納されて返ります。

AreaType areaType

- ・ enum の AreaType を指定します。  
Office, Living, Museum のいずれかを設定できます。
- ・ 省略可能です。省略時は Office が設定されます。

【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。
NotReadFileData	データ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。
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明るさ画像データ(NB 値)と指定された範囲の明るさ感推定値が返ります。

**EsrtimatingBR** クラスに明るさ感推定値を求める範囲をまとめて配列で指定することにより、その範囲の明るさ感推定値と各 NB 値範囲の割合が書き込まれて返ります。

【構文】

```
enmResult getEstimationValuesFromOXYZ(string oxyz_Filename,
ref double[,] NB, ref EstimatingBR[] data, AreaType areaType = AreaType.Office)
```

【引数】

string oxyz\_Filename

- ・ OXYZ ファイル名です。

ref double[,] NB

- ・ 参照引数です。明るさ NB 値が格納されて返ります。

ref EstimatingBR[] data

- ・ 参照引数です。明るさ感推定値の測定範囲を指定した **EsrtimatingBR** クラスの配列です。

ここに明るさ感推定値と各明るさ検討用データの割合が格納されて返ります。

**AreaType areaType**

- ・ enum の AreaType を指定します。  
Office, Living, Museum のいずれかを設定できます。
- ・ 省略可能です。省略時は Office が設定されます。

**【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。
NotReadFileData	データ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。
NullStatus	メソッドの状態異常が発生しました。
UnknownError	不明なエラー
Ranges_Are_Wrong	最大輝度か圧縮率の指定値が間違っています。

### 3 getIlluminance() メソッド

輝度ファイルや輝度データと反射率から求めた照度画像データ(lx)が返ります。

3-1 輝度ファイルと反射率から求めた照度画像データ(lx)が返ります。

**【構文】**

enmResult getIlluminance(string LM\_Filename , double Reflectance , ref double[,] illuminance)

**【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  
LM ファイルのデータ形式については III データ形式 (20 ページ) を参照してください。

double Reflectance

- ・ 反射率の値

ref double[,] illuminance

- ・ 参照引数です。照度画像データ(lx)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。
-----------------	------------------------

3-2 輝度データと反射率から求めた照度画像データ(lx)が戻ります。

**【構文】**

enmResult getIlluminance(float[,] luminance , double Reflectance , ref double[,] illuminance)

**【引数】**

float[,] luminance

- ・ 輝度画像データ配列です。

double Reflectance

- ・ 反射率の値

ref double[,] illuminance

- ・ 参照引数です。照度画像データ(lx)が格納されます。

**【戻り値】**

返却値	意味
Success	成功しました。
NoLuminanceData	輝度画像データがありません。

## 4 getIlluminanceFromOXYZ() メソッド

OXYZ ファイルの輝度データと反射率から求めた照度画像データ(lx)が戻ります。

4-1 OXYZ ファイルの輝度データと反射率から求めた照度画像データ(lx)が戻ります。

**【構文】**

enmResult getIlluminanceFromOXYZ(  
string oxyz\_Filename ,double Reflectance , ref double[,] illuminance)

**【引数】**

string oxyz\_Filename

- ・ OXYZ ファイル名です。

double Reflectance

- ・ 反射率の値

ref double[,] illuminance

- ・ 参照引数です。照度画像データ(lx)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。
NotReadOxyzHeader	OXYZ ファイルからのヘッダが読み込めませんでした。

## 5 getRealAppearance() メソッド

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。sRGB とはガンマ 2.2 補正を掛けた 0~255 の RGB の値です。 (5-1~5-2)

5-1 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。

【構文】

```
enmResult getRealAppearance(string LM_Filename ,
double monitorMaxLuminance , double compressionRatio , ref byte[ ][,] sRGB)
```

【引数】

string LM\_Filename

- ・ LM ファイル名です。  
LM ファイルのデータ形式については III データ形式 (23 ページ) を参照してください。

double monitorMaxLuminance

- ・ モニタの最大輝度値

double compressionRatio

- ・ 圧縮率の値

ref 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。
NullStatus	メソッドの状態異常が発生しました。
UnknownError	不明なエラー

5-2 輝度画像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。

【構文】

```
enmResult getRealAppearance(float[,] luminance ,
double monitorMaxLuminance , double compressionRatio , ref byte[ ][ ] sRGB)
```

【引数】

- float[,] luminance
- ・ 輝度画像データ配列です。
- double monitorMaxLuminance
- ・ モニタの最大輝度値
- double compressionRatio
- ・ 圧縮率の値
- ref 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。
NoLuminanceData	輝度データが読み込めませんでした。
NullStatus	メソッドの状態異常が発生しました。
UnknownError	不明なエラー

## 6 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。

【構文】

```
enmResult getRealAppearanceFromOXYZ(string oxyz_Filename ,
double monitorMaxLuminance , double compressionRatio , ref byte[ ][ ] sRGB)
```

【引数】

- string oxyz\_Filename
- ・ OXYZ ファイル名
- double monitorMaxLuminance
- ・ モニタの最大輝度値
- double compressionRatio
- ・ 圧縮率の値

ref 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。
NullStatus	メソッドの状態異常が発生しました。
UnknownError	不明なエラー

## 7 savePictureLuminance() メソッド

輝度画像データから輝度画像を生成して PNG ファイルで保存します。

【構文】

```
enmResult savePictureLuminance(string savePNGfilename ,
float[,] luminance , bool WithColorBar , double min , double max)
```

【引数】

string savePNGfilename

- ・ 保存する PNG ファイル名

float[,] luminance

- ・ 輝度画像データ配列

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。

## 8 savePictureNB() メソッド

明るさ画像データ(NB)から明るさ画像と明るさ検討画像を生成して PNG ファイルで保存します。

### 【構文】

```
enmResult savePictureNB(string savePNGfilename ,
double[,] NB , bool WithColorBar , double min , double max)
```

### 【引数】

string savePNGfilename

- ・ 保存する PNG ファイル名

double[,] NB

- ・ 明るさ(NB)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。

## 9 savePictureRealAppearance() メソッド

sRGB 画像データからリアルアピア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。

## 10 savePictureIlluminance() メソッド

照度画像データから照度画像を生成して PNG ファイルで保存します。

【構文】

```
enmResult savePictureIlluminance(
string savePNGfilename ,double[,] illuminance , bool WithColorBar , double min , double max)
```

【引数】

string savePNGfilename

- ・ 保存する PNG ファイル名

double[,] illuminance

- ・ 明るさ(NB)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。

## 11 getOxyzFromJpegs() メソッド

1 枚以上の Jpeg 画像から XYZ データを求めて OXYZ ファイルを保存します。

### 【構文】

```
enmResult getOxyzFromJpegs(JpegParam jpegParam , string oxyz_Filename)
```

### 【引数】

JpegParam jpegParam

- ・ Jpeg 画像を合成するときのパラメータデータ

string oxyz\_Filename

- ・ 保存する OXYZ データのファイル名

### 【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ない。
OxyzFileSaveError	OXYZ ファイルを保存できませんでした。
NullStatus	メソッドの状態異常が発生しました。
UnknownError	不明なエラー

## 12 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 EstimatingBR クラス

明るさ感推定値を求めるために利用します。

明るさ感推定値を求める範囲とその範囲より求めた明るさ感推定値、および明るさ感を上げる領域、明るくも暗くもない領域、明るさ感を下げる領域が占める割合が保持されます。

### 2.1 プロパティ

プロパティ一覧

	名前	型	説明
1	PresBrValue	Double	求めた明るさ感推定値を保持しています。
2	DarkPercent	Double	求めた明るさ感を下げる領域が占める割合を保持しています。
3	NutralPercent	Double	求めた明るくも暗くもない領域が占める割合を保持しています。
4	BrightPercent	Double	求めた明るさ感を上げる領域が占める割合を保持しています。
5	area	Area	EstimatingBR クラス内に定義された Area クラスによって定義された明るさ感推定値を求める範囲を保持します。

### 2.2 サブクラス Area クラス

EstimatingBR クラス内に定義された明るさ感推定値を求める範囲が保持されるクラスです。

Area クラスのプロパティ一覧

	名前	型	説明
1	Left	Int	明るさ感推定値を求める範囲の左上の X 座標を設定します。
2	Top	Int	明るさ感推定値を求める範囲の左上の Y 座標を設定します。
3	Width	Int	明るさ感推定値を求める範囲の幅を設定します。
4	Height	Int	明るさ感推定値を求める範囲の高さを設定します。

## 2.3 使用例

スクリプト例

```
Omni omni1 = new Omni();
EstimatingBR ebr = new EstimatingBR(new EstimatingBR.Area( 29, 129, 160, 186));
double[,] nb = null;
enmResult result = omni1.getEstimationValues(@"C:\¥temp¥Sample.lm", ref nb, ref ebr);
Print(ebr.PresBrValue.ToString());
Print(ebr.BrightPercent.ToString());
Print(ebr.NutralPercent.ToString());
Print(ebr.DarkPercent.ToString());
```

上記の結果表示

```
242.6
44.39
13.31
5.06
```

REALAPS-Omni ソフトでの表示例

画面最大輝度 200.0 圧縮係数 0.500 再表示

最大輝度 43764.09 cd/m<sup>2</sup> オーバーフロー R 4.0338 G 5.1678 B 8.7670

明るさ感推定値 242.60 明るさ感を上げる領域 44.39% 明るくも暗くもない領域 13.31% 明るさ感を下げる領域 5.06%

測定枠表示 計測  計測結果を画像に含める 画像の保存

左上座標 X 29 Y 129 幅 160 高さ 186

### 3 JpegParam クラス

Jpeg 画像から XYZ 画像データを求めるために利用します。

合成する Jpeg 画像のファイル名、画像の水平画角、垂直画角の設定をおこないます。また、画像タイプが全方位画像か透視投影画像かの設定もおこないます。

#### 3.1 プロパティ

プロパティ一覧

	名前	型	説明
1	JpegFilesnames	String[]	Jpeg 画像ファイル名を格納する配列です。
2	HorizontalViewAngle	Double	水平画角を設定します。
3	VerticalViewAngle	Double	垂直画角を設定します。
4	Projection	ProjectionType	JpegParam クラス内に定義された enum ProjectionType によって定義された画像のタイプを設定します。

enum ProjectionType 一覧

	メンバー名	説明
1	Perspective = 0,	透視投影画像です。
2	Omni = 1,	全方位画像です。

#### 3.2 使用例

設定の例

```
Omni omni1 = new Omni();
JpegParam jpegParam = new JpegParam();
jpegParam.JpegFilesnames = new string[6];
jpegParam.JpegFilesnames[0] = @"C:\temp\用賀駅ビルエリア 13¥R0010659.JPG";
jpegParam.JpegFilesnames[1] = @"C:\temp\用賀駅ビルエリア 13¥R0010660.JPG";
jpegParam.JpegFilesnames[2] = @"C:\temp\用賀駅ビルエリア 13¥R0010661.JPG";
jpegParam.JpegFilesnames[3] = @"C:\temp\用賀駅ビルエリア 13¥R0010662.JPG";
jpegParam.JpegFilesnames[4] = @"C:\temp\用賀駅ビルエリア 13¥R0010663.JPG";
jpegParam.JpegFilesnames[5] = @"C:\temp\用賀駅ビルエリア 13¥R0010664.JPG";
jpegParam.HorizontalViewAngle = 360.0;
jpegParam.VerticalViewAngle = 180.0;
jpegParam.Projection = JpegParam.ProjectionType.Omni;
enmResult result = omni1.getOxyzFromJpegs(jpegParam,
@"C:\temp\用賀駅ビルエリア 13¥Sample.oxyz");
```

### 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。

