

# NaturalPoint

## OptiTrack API

### Version: 1.7

### For SDK Version 1.1.032

### Date: 08/21/2007

#### **NaturalPoint Proprietary**

本ドキュメントに含まれ、または開示されているデータと情報は、**NaturalPoint Corporation** に帰属する秘密情報であり、すべての著作権を明示的に留保しています。受領者は、本ドキュメントの受領により、このドキュメントおよび掲載されている情報が秘密情報として保持されており、**NaturalPoint Corporation** の書面による同意なしに、その手段を問わず、すべてまたは一部を使用、コピー、複製できないことに同意したものとします。本ドキュメントの情報は、準備段階にあり、変更される場合があります。また、**NaturalPoint Corporation** は、本ドキュメントの内容について一切の責任を負いません。



**NaturalPoint Corporation**

33872 SE Eastgate Circle

Corvallis OR 97339

Copyright © 2004–2007 **NaturalPoint Corporation**. All rights reserved.

Printed in the US.

## もくじ

もくじ	3
1 アーキテクチャ	5
1.1 概要	5
1.2 コンポーネントモデル	5
1.3 インタフェースレイアウト	5
2 機能	7
2.1 nusb.sys	7
2.2 cameradll.dll	7
2.3 optitrack.dll	7
3 設計上の留意点	8
3.1 はじめに	8
3.2 シングルドットトラッキング	10
3.3 ベクトルトラッキング	10
3.4 コネクションポイント	11
3.5 スレッディングの問題	11
3.6 スムージング	11
3.7 カメラコマンド	11
3.8 カメラフレームオブジェクトの寿命	12
3.9 オブジェクト座標	12
3.10 カラーストラクチャ	13
3.11 VARIANT_BOOL と BOOL	13
3.12 スイッチの状態	13
4 インタフェース	14
4.1.1 INPCameraCollection	14
4.1.1.1 プロパティ	14
4.1.1.2 メソッド	15
4.1.2 INPCamera	16
4.1.2.1 プロパティ	16
4.1.2.2 メソッド	19
4.1.3 INPVector	34
4.1.3.1 プロパティ	34
4.1.3.2 メソッド	36
4.1.4 INPVector2	37
4.1.4.1 Properties	38
4.1.4.2 メソッド	41
4.1.5 INPVector3	41
4.1.6 INPPoint	45
4.1.6.1 プロパティ	45
4.1.7 INPSmoothing	46
4.1.7.1 プロパティ	46
4.1.7.2 メソッド	48
4.1.8 INPCameraFrame	48
4.1.8.1 プロパティ	49
4.1.8.2 メソッド	52
4.1.9 INPObject	53

4.1.9.1	プロパティ	53
4.1.9.2	メソッド	55
4.1.10	INPAvi	55
4.1.10.1	プロパティ	56
4.1.10.2	メソッド	57
4.1.11	_INPCameraCollectionEvents	58
4.1.11.1	メソッド	58
4.1.12	_INPCameraEvents	59
4.1.12.1	メソッド	59
5	サンプルコード	60
5.1	VBScript	60
5.2	VB.NET 2003	60
5.3	VC6	60

# 1 アーキテクチャ

## 1.1 概要

OptiTrack API は、一連の COM オートメーションインタフェースとして記述されています。このタイプのインタフェースを選択したのは、高い柔軟性を備えているという理由からです。COM オートメーションインタフェースは、VBScript、JavaScript、Visual Basic、C/C++を含むほぼすべての言語でサポートされています。Python、Delphi など、その他多くの言語もサポートしています。詳細は任意の言語に関する記述を参照してください。

## 1.2 コンポーネントモデル

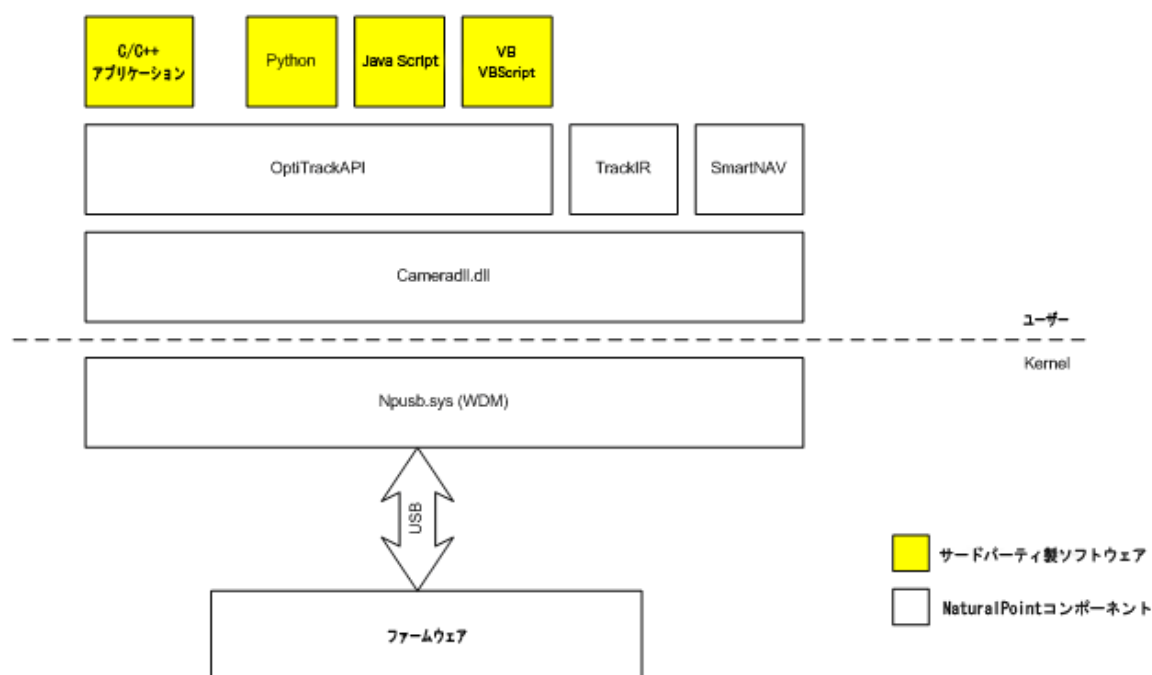


図 1

## 1.3 インタフェースレイアウト

図 2 に、OptiTrack API のインタフェースおよびそれらの関係を示します。メインのインタフェースは INPCameraCollection です。このインタフェースは、システムに接続されているすべてのカメラのリストを保持しています。システムにカメラを追加したり、取り除いたりすると、\_INPCameraCollectionEvents インタフェースによって通知が送信されます。

INPCamera インタフェースには、デバイスに関連するほとんどの機能が含まれています。デバイスに関する情報の問い合わせ、デバイスの起動と停止、データの収集が可能です。

デバイスからフレームがキャプチャされる際に、クライアントは INPCameraFrame インタフェースを取得します。このインタフェースは、フレームでトラッキングされたオブジェクトの単なる集合です。フレームにおけるオブジェクトのランク付けは、INPCamera インタフェースでオプションを設定することによって変更できます。カメラを含むイベントに関する通知は、\_INPCameraEvents インタフェースを通じて受け取ります。

INPSmoothing と INPVector インタフェースは、フレームデータを操作するためのヘルパー関数として提供されています。カメラからのデータには空間雑音が含まれる場合があるので、ある種のスムージングを適用する必要があります。INPSmoothing インタフェースは、スムージングアルゴリズムの一例です。単一のドットオブジェクトの位置やベクトルの演算に対して、スムージングを適用してください。

Vector Expansion を使用する場合、INPVector インタフェースは 6 つのパラメータの値を計算します。その計算を正しく行うためには、Vector クリップが必要です。Vector の位置を計算する際は、出力に対して必ず何らかのスムージングを適用してください。INPVector インタフェースへのフレーム入力に対しては、スムージングは必要ありません。

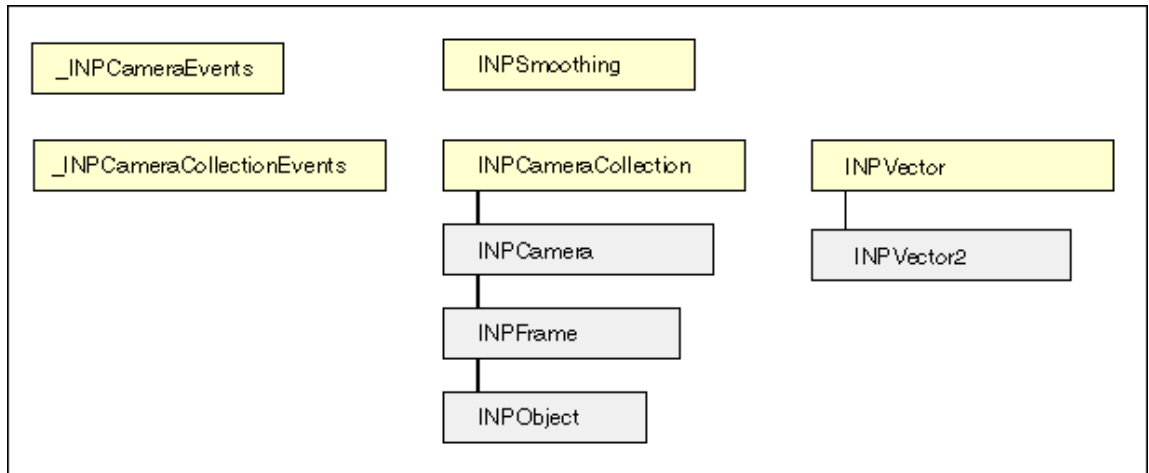


図 2

## 2 機能

### 2.1 nusb.sys

nusb.sys は、NaturalPoint カメラの USB ドライバです。ドライバはカーネル空間に常駐し、システムにデバイスが接続されるたびにロードされます。ドライバは、他の NaturalPoint コンポーネントのデバイスとのコミュニケーションを処理します。デバイスドライバとの対話は不要です。

### 2.2 cameradll.dll

cameradll はユーザー空間に常駐し、USB ドライバとの通信を処理します。このライブラリには、カメラ特有のデータを処理するルーチンが含まれています。外部インターフェースは提供されていません。

### 2.3 optitrack.dll

この DLL は、OptiTrack が提供する一連の COM オートメーションインターフェースです。

## 3 設計上の留意点

### 3.1 はじめに

ほんの少量のコードで、カメラと通信できるようになります。カメラの初期化手順を以下に概説します。

1. INPCameraCollection オブジェクトを作成します。
2. システム上のデバイスを列挙するために、Enum() メソッドを呼び出します。
3. 収集したデバイスからカメラを探します。
4. カメラが検出された場合は、カメラを初期化します。
  - a. INPCamera の Open() メソッドを呼び出します。
  - b. ハードウェアのタイプ、シリアル番号など、デバイスに関する情報を問い合わせます（必要な場合）。
  - c. お使いのアプリケーションに関連があるオプションを適用します（必要な場合）。
  - d. LED を点灯します（必要な場合）。
  - e. Start() メソッドを呼び出します。これによって、カメラからのフレーム情報の収集が開始されます。

この時点で、カメラは初期化され、フレーム情報を収集しているはずですが、アプリケーションのメインループでは、\_INPCameraEvents インタフェースを使ってフレームコールバックを処理するか、INPCamera の GetFrame() を使ってフレームをポーリングする必要があります。

データ処理が完了したら、INPCamera の Stop() メソッドを呼び出し、その後 Close() を呼び出してデバイスをクローズします。

以下のコードで、VBScript のコールバックを使用してカメラと通信する方法を示します。

```
dim objNPCameras
dim objCamera

Sub SINK_DeviceArrival(objCamera)
    WScript.Echo "Device Arrival."
End Sub

Sub SINK_DeviceRemoval(objCamera)
    WScript.Echo "Device Removal."
End Sub

Sub CAMERA_FrameAvailable(objCamera)
    Set objFrame = objCamera.GetFrame(0)

    ' TODO: process the frame information here

    objFrame.Free()
End Sub

' create the main config object
set objNPCameras = WScript.CreateObject("OptiTrack.NPCameraCollection", "SINK_")

' enumerate the cameras
objNPCameras.Enum()
WScript.Echo "Num of cameras: " & objNPCameras.Count

for each objCamera in objNPCameras

    ' register callbacks
    WScript.ConnectObject objCamera, "CAMERA_"

    ' Open the camera
    objCamera.Open()

    ' start the camera
    objCamera.Start()

    ' process data until stopped
    WScript.Echo "hit OK to stop data."

    ' close the camera
    objCamera.Close()

    ' disconnect callbacks
    WScript.DisconnectObject objCamera

Next
```

## 3.2 シングルドットトラッキング

基本的な OptiTrack API は、カメラのビューにおける単一のオブジェクトをトラッキングする際に役立つ機能を提供しています。どのオブジェクトをトラッキング対象のオブジェクトとするかを定める際に、いくつかの要因が考慮されます。これらのオプションは、INPCamera インタフェースの SetOption メソッドによって公開されます。

トラッキングするオブジェクトの決定には、6 つの変数が関与しています。オブジェクトのサイズ、オブジェクトの比率、他のオブジェクトへの近接度、静止カウント（不動性）、イメージャの中心からの距離、現在のオブジェクトが最後にトラッキングしたオブジェクトかどうか、の 6 つです。

オブジェクトのサイズは、オブジェクトに含まれるイメージャでのピクセル数です。比率は、オブジェクトの高さに対する幅の比率です。静止カウントは、オブジェクトが動いているかどうかの指標です。多くの場合、静止しているオブジェクトはトラッキングには適していません。中心からの距離は、オブジェクトがイメージャの中心からどのくらい離れているかの指標です。イメージャの中心から離れるほど、トラッキングオブジェクトとしては不適切な可能性が高くなります。最後にトラッキングしたオブジェクトかどうかは、計算においてより重要とみなされます。これは、直前にトラッキングしたオブジェクトは、今後もトラッキングされる可能性も高いという事実に基づいています。

トラッキングに関するすべての変数には、最小、最大、理想、重みがあります。もし、この値が最小と最大からはずれた場合のペナルティがあります。6 つの各変数は、最小と最大の範囲における値の位置に基づいて、0 から 1.0 の値に調整されます。調整された値に変数の重みを掛け合わせた後、すべての変数の合計を計算します。そして、オブジェクトはその重み付けに基づいてランク付けされます。最高の重み付けをされたオブジェクトは第 1 位にランク付けされます。

オプションの変更は、INPCameraFrame インタフェースで返されるオブジェクトのランクに影響を与えます。最高ランクのオブジェクトは、トラッキング対象のオブジェクトとなります。単一のオブジェクトのみを選択する場合には、動きを識別する目的で、アプリケーションはこのオブジェクトを使用すべきです。

カメラからの出力は空間雑音を含んでいます。シングルドットトラッキングを使用する際は、適切なアルゴリズムによって必ずデータをスムージングしてください。INPSmoothing インタフェースでは、スムージングアルゴリズムのサンプルが提供されています。

OptiTrack API が提供するドットトラッキング機能は、すべてオプションです。API で提供するアルゴリズムは、他の NaturalPoint アプリケーションが使用するルールセットと同一です。サードパーティ製アプリケーションで、デフォルトのトラッキングパラメータでは期待した結果が得られない場合は、カメラオブジェクト情報の取り込みとオブジェクトのランク付けを手動で行ってください。

## 3.3 ベクトルトラッキング

ベクトルトラッキングは、3D 空間におけるユーザの頭部の位置と方向を決定するために使用できるオプションです。シングルドットトラッキングのランク付けは、ベクトルトラッキングアルゴリズムには影響を与えません。INPVector インタフェースは、それが有効と判断するオブジェクトをランク付けします。この判断は、オブジェクトの位置とサイズに基づいて行われます。ベクトルトラッキングを使用する際は、視野内の余分なオブジェクトを最低限に抑えることが重要です。

何らかの理由で Vector クリップの 3 つすべてのオブジェクトのトラッキングができなくなった場合は、INPVector の Reset メソッドを呼び出して計算を再開します。

ベクトルアルゴリズムは、クリップがカメラの前方、約 2 から 3 フィート離れて置かれている場合に最も効果的です。

### 3.4 コネクションポイント

コールバック通知には、標準の COM コネクションポイントが使用されます。コネクションポイントを選択した理由は、COM オートメーションインタフェースと通信する、ほとんどの言語と互換性があるからです。詳しくは、MSDN ヘルプの COM 関連文書を参照してください。特に、IConnectionPoint と IConnectionPointContainer をお読みください。

C/C++ではコネクションポイントの設定にやや手間がかかりますが、コーディングを支援するためにサンプルコードが提供されています。

### 3.5 スレディングの問題

OptiTrack API は、アパートメントスレッドです。アパートメントスレッドとは、任意の時間に DLL の呼び出しを実行するためには、ただ 1 つのスレッドしか使用できないことを意味します。あるスレッドが DLL で実行中の場合には、その DLL への呼び出しに使われる他のすべてのスレッドは、最初のスレッドが完了するまでブロックされます。

これはコールバックの実装方法にも影響を与えます。コネクションポイントのコールバックは、オブジェクトを作成した同じスレッドで行う必要があります。コネクションポイントを持つオブジェクトの作成時に、隠しウィンドウを作成することによって、これを実行することができます。隠しウィンドウは、呼び出しスレッドのメッセージキューを使います。コールバックは、隠しウィンドウへメッセージをポストして、コネクションポイントコールバックインタフェースを呼び出すことによって実行されます。

このメカニズムの欠点は、OptiTrack 通知がタイムリーにハンドルされる場合には、これを実行するために、オブジェクトを作成したスレッドに対するメッセージキューを解放する必要がありますということです。たとえば、OptiTrack カメラオブジェクトを作成するために GUI アプリケーションのメインのスレッドを使う場合は、そのスレッドでブロック操作が実行されていないことを確実にします。ブロック操作が必要な場合は、メッセージの処理にはメッセージループを使用します。

この問題は、コネクションポイントコールバックにのみ影響を及ぼします。この INPCamera インタフェースのポーリングによってカメラフレーム情報を収集する場合には、これは問題にはなりません。

### 3.6 スムージング

INPSmoothing インタフェースは、X と Y の値の間の相関を仮定します。このインタフェースは、通常、シングルドットトラッキングのデータのスムージングとフィルタリングに使われます。スムージングのアルゴリズムは、単一の値をスムージングする場合（たとえばベクトル計算から算出されたヨーのスムージング）にも使用できます。この場合は、X の値のみを使用し、Y の値を 0 に設定します。

### 3.7 カメラコマンド

カメラデバイスへのコマンドを頻繁に発行すると、カメラのビデオスループットのパフォーマンスが低下する可能性があります。物理的なカメラと対話するすべてのコマンド

は、デバイスに割り込みを発生させ、データ収集と USB 伝送レートを遅らせる原因となります。通常の動作中は、デバイスはトラッキング情報のキャプチャの処理で手一杯です。クライアントからのコマンドが頻繁に発行されると、データフローが中断され、期待した結果が得られない場合もあります。

データをキャプチャする際は、LED コマンドの使用を制限してください。状態を示すために定期的に LED をオン/オフすることは構いませんが、LED の点滅はお勧めしません。さらに、LED の状態をユーザのアプリケーションで内部的に監視して、カメラコマンドは状態変化の際にだけ発光されるようにすることを推奨します。

### 3.8 カメラフレームオブジェクトの寿命

OptiTrack システムにおいて、カメラフレームオブジェクトは限られたリソースです。必ず `Free()` メソッドを呼び出し、できるだけ早く `INPCameraFrame` インタフェースを解放してください。

### 3.9 オブジェクト座標

デバイスが通常の直立姿勢にあれば、オブジェクト座標はデバイスの左上隅から計測されます。図 3 にデフォルトの設定を示します。



図 3

変換が適用されると、ドットの位置がわずかに変化します。変換された座標はデバイスの中心から相対的に計測されます。変換には、カメラの方向と X 軸と Y 軸のミラーリングが含まれます。変換は、`INPObject` インタフェースの `Transform` メソッドを呼び出すことによって、実行します。図 4 に、新しい設定を示します。

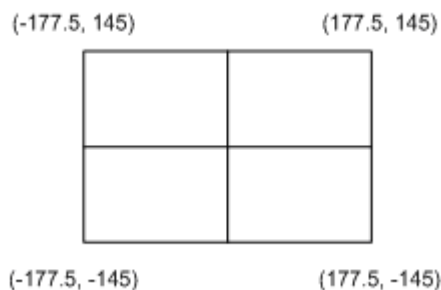


図 4

### 3.10 カラーストラクチャ

Visual Basic におけるカラーストラクチャの処理は、Win32 API とは異なります。Win32 API では、ほとんどの色の指定に対して RGB マクロが使われます。Visual Basic では、独自の Color オブジェクトを使って色を処理します。これら 2 つの間に直接的な互換性はありません。

OptiTrack API では、RGB マクロを使って色を指定する必要があります。Visual Basic のサンプルコードでは、変換に役立つ 2 つの構造体が提供されています。ARGBColor は Visual Basic の色構造体であり、NPColor は OptiTrack API へ渡される色構造体を表現することができます。

### 3.11 VARIANT\_BOOL と BOOL

アプリケーションの作成で C/C++ を使用する開発者が注意すべきことの 1 つに、VARIANT\_BOOL と BOOL は全く同じというわけではないということがあります。VARIANT\_BOOL では、true を示す戻り値として -1 が定義されています。コードの書き方によっては、このことがアプリケーションに影響を与える可能性があります。

この状況を解決するために、いくつかのマクロが提供されています。VARIANT\_BOOL を比較する際には、VARIANT\_TRUE と VARIANT\_FALSE を使用してください。また、OptiTrack API では、VARIANT\_BOOL への（からの）変換のために、2 つのマクロ (B2VB と VB2B) が提供されています。

### 3.12 スイッチの状態

各フレームをキャプチャする際、外部スイッチの状態をソフトウェアスタックが利用することができます。スイッチの状態については、INPCameraFrame インタフェースの SwitchState プロパティに対して問い合わせます。

デフォルトでは、システムでのデータフローを低減するために、OptiTrack API は空のカメラフレームを省略します。この動作によって、トラッキングオブジェクトが可視でない場合には、スイッチの状態の情報も失われることとなります。すべてのフレームがクライアントに送られるようにするためには、NP\_OPTION\_SEND\_EMPTY\_FRAMES を true に設定してください。

注：OptiTrack 3 Flex ハードウェアでは、外部スイッチはプラグで外へ取り出すことはできません。

## 4 インタフェース

### 4.1.1 INPCameraCollection

これは、カメラと通信するためのメインのエントリポイントです。このインタフェースは、システムでの利用可能なすべてのカメラのリストを保持する、標準的な COM の列挙型インタフェースです。

#### 4.1.1.1 プロパティ

##### 4.1.1.1.1 INPCameraCollection::get\_\_NewEnum

このプロパティは列挙子のコピーを返します。返されるオブジェクトは、コレクションの最初のオブジェクトをポイントするインデックスを保持する、INPCamera インタフェースです。

```
HRESULT get__NewEnum(LPUNKNOWN *ppunk);
```

パラメータ

*ppunk*

[out, retval] IUnknown インタフェースへのポインタです。

戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

##### 4.1.1.1.2 INPCameraCollection::get\_Count

コレクションのオブジェクトの数を返します。

```
HRESULT get_Count(LONG *pVal);
```

パラメータ

*pVal*

[out, retval] カウントを受け取る LONG 型のポインタです。

戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

## 4.1.1.2 メソッド

### 4.1.1.2.1 INPCameraCollection::Item

Item メソッドは、コレクションの中の対応するアイテムを返します。この例では、Item は INPCamera インタフェースを返します。

```
HRESULT Item(LONG a_vlIndex, INPCamera **ppCamera);
```

#### パラメータ

*A\_vlIndex*

[in] 取得するアイテムのインデックスです。

*ppCamera*

[out, retval] INPCamera インタフェースを受け取るポインタです。

#### 戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

### 4.1.1.2.2 INPCameraCollection::Enum

Enum メソッドは、システムにおけるすべてのカメラを列挙します。このメソッドは、INPCameraCollection インタフェースの関数の中で、最初に呼び出す必要があります。

```
HRESULT Enum();
```

#### パラメータ

なし

#### 戻り値

値	意味
S_OK	メソッドは成功

### 4.1.1.2.3 INPCameraCollection::Synchronize

Synchronize メソッドは、複数のカメラの露出タイミングとフレーム ID をシンクします。このメソッドは、OptiTrack FLEX:C120 と V100 でのみ有効で、動作させるためには、最初にカメラ間をシンクケーブルで接続する必要があります。このメソッドが呼び出されると、システムのすべてのカメラが停止し、マスターが選ばれ、その後、すべてのカメラが起動されます。

```
HRESULT Synchronize();
```

#### パラメータ

なし

#### 戻り値

値	意味
S_OK	メソッドは成功

## 4.1.2 INPCamera

### 4.1.2.1 プロパティ

#### 4.1.2.1.1 INPCamera::get\_SerialNumber

読み取り専用。カメラのシリアル番号です。

```
HRESULT get_SerialNumber(LONG *pVal);
```

パラメータ

*pVal*

[out, retval] カメラのシリアル番号を受け取る LONG 型のポインタです。

戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

#### 4.1.2.1.2 INPCamera::get\_Model

読み取り専用。カメラのモデルです。有効な値は、以下の表のとおりです。

名前	説明
NP_HW_MODEL_OLDTRACKIR	旧型 TrackIR 1
NP_HW_MODEL_SMARTNAV	SmartNAV
NP_HW_MODEL_TRACKIR	TrackIR2 または TrackIR3
NP_HW_MODEL_OPTITRACK	OptiTrack
NP_HW_MODEL_UNKNOWN	不明なハードウェア

```
HRESULT get_Model(LONG *pVal);
```

パラメータ

*pVal*

[out, retval] モデルを受け取る LONG 型のポインタです。

戻り値

値	意味
S_OK	メソッドは成功

E\_POINTER

ポインタが無効

**4.1.2.1.3 INPCamera::get\_Revision**

読み取り専用。カメラのハードウェアリビジョンです。リビジョンは、カメラのモデルによって異なります。以下の表に、有効なリビジョンの値を示します。

名前	説明
NP_HW_REVISION_OLDTRACKIR_LEGACY	旧型 TrackIR / SmartNAV
NP_HW_REVISION_OLDTRACKIR_BASIC	旧型 TrackIR / SmartNAV
NP_HW_REVISION_OLDTRACKIR_EG	旧型 TrackIR / SmartNAV
NP_HW_REVISION_OLDTRACKIR_AT	旧型 TrackIR / SmartNAV
NP_HW_REVISION_OLDTRACKIR_GX	旧型 TrackIR / SmartNAV
NP_HW_REVISION_OLDTRACKIR_MAC	旧型 TrackIR / SmartNAV
NP_HW_REVISION_SMARTNAV_BASIC	SmartNAV
NP_HW_REVISION_SMARTNAV_EG	SmartNAV
NP_HW_REVISION_SMARTNAV_AT	SmartNAV
NP_HW_REVISION_SMARTNAV_MAC_BASIC	SmartNAV
NP_HW_REVISION_SMARTNAV_MAC_AT	SmartNAV
NP_HW_REVISION_TRACKIR_BASIC	TrackIR
NP_HW_REVISION_TRACKIR_PRO	TrackIR
NP_HW_REVISION_OPTITRACK_BASIC	OptiTrack Basic
NP_HW_REVISION_OPTITRACK_FLEX	OptiTrack Flex
NP_HW_REVISION_UNKNOWN	不明

```
HRESULT get_Revision(LONG *pVal);
```

**パラメータ***pVal*

[out, retval] リビジョンを受け取る LONG 型のポインタです。

**戻り値**

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

#### 4.1.2.1.4 INPCamera::get\_Width

イメージの幅をピクセルで返します。

```
HRESULT get_Width(LONG *pVal);
```

パラメータ

*pVal*

[out, retval] イメージの幅を受け取る LONG 型のポインタです。

戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

#### 4.1.2.1.5 INPCamera::get\_Height

イメージの高さをピクセルで返します。

```
HRESULT get_Height(LONG *pVal);
```

パラメータ

*pVal*

[out, retval] 高さを受け取る LONG 型のポインタです。

戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

#### 4.1.2.1.6 INPCamera::get\_FrameRate

イメージのフレームレートを1秒あたりのフレーム数で返します。

```
HRESULT get_FrameRate(LONG *pVal);
```

パラメータ

*pVal*

[out, retval] フレームレートを受け取る LONG 型のポインタです。

戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

## 4.1.2.2 メソッド

### 4.1.2.2.1 INPCamera::Open

デバイスへ接続します。このメソッドは、INPCamera インタフェースの呼び出しの中で、最初に呼び出す必要があります。

**HRESULT** Open();

**パラメータ**

なし

**戻り値**

**値**

S\_OK

NP\_E\_DEVICE\_DISCONNECTED

E\_POINTER

**意味**

メソッドは成功

デバイスがシステムから取り除かれている

ポインタが無効

### 4.1.2.2.2 INPCamera::Close

デバイスとの接続を切断します。

**HRESULT** Close();

**パラメータ**

なし

**戻り値**

**値**

S\_OK

**意味**

メソッドは成功

### 4.1.2.2.3 INPCamera::Start

カメラでビデオを開始します。Start が呼び出されると、データフレームが利用可能になります。

**HRESULT** Start();

**パラメータ**

なし

**戻り値**

**値**

S\_OK

**意味**

メソッドは成功

NP_E_DEVICE_DISCONNECTED	デバイスがシステムから取り除かれている
NP_E_DEVICE_NOT_SUPPORTED	接続されているデバイスは、この API でサポートされない古いタイプのハードウェアである

#### 4.1.2.2.4 INPCamera::Stop

カメラからのデータフローを停止します。

```
HRESULT Stop();
```

**パラメータ**

なし

**戻り値**

**値**

S\_OK

NP\_E\_DEVICE\_NOT\_SUPPORTED

**意味**

メソッドは成功

接続されているデバイスは、この API でサポートされない古いタイプのハードウェアである

#### 4.1.2.2.5 INPCamera::SetLED

指定の LED の状態を設定します。このメソッドで、LED をオンまたはオフに設定できます。

```
HRESULT SetLED(LONG lLED, VARIANT_BOOL fOn);
```

**パラメータ**

*lLED*

[in] オンまたはオフにする、LED です。LED は NP\_LED 列挙子によって指定しなければなりません。LED には、ハードウェアリビジョンによって異なるマッピングがあります。以下の表に、その違いをまとめています。

**TrackIR / SmartNAV / OptiTrack Basic**

**LED**

NP\_LED\_ONE

NP\_LED\_TWO

NP\_LED\_THREE

NP\_LED\_FOUR

**意味**

イルミネーション LED

緑色 LED (状態)

赤色 LED

青色 LED

**OptiTrack Flex**

LED	意味
NP_LED_ONE	イルミネーション LED
NP_LED_TWO	左赤色 LED (状態)
NP_LED_THREE	右赤色 LED
NP_LED_FOUR	使用されていない

*fOn*

[in] 設定する LED の状態です。VARIANT\_TRUE で LED はオンに、VARIANT\_FALSE で LED はオフになります。

#### 戻り値

値	意味
S_OK	メソッドは成功
E_INVALIDARG	無効な LED 番号
NP_E_DEVICE_DISCONNECTED	デバイスがシステムから取り除かれている
NP_E_DEVICE_NOT_SUPPORTED	接続されているデバイスは、この API でサポートされない古いタイプのハードウェアである

#### 4.1.2.2.6 INPCamera::SetVideo

カメラからのビデオストリームをオンまたはオフに設定します。これによって、カメラを停止せずにビデオを一時停止することができます。カメラは、このメソッドを呼び出す前に起動しておく必要があります。INPCamera::Stop() を呼び出すと、その時点の状態にかかわらず、ビデオストリームが強制的にオフになります。

```
HRESULT SetVideo(VARIANT_BOOL fOn);
```

#### パラメータ

*fOn*

[in] 設定するビデオストリームの状態です。VARIANT\_TRUE はビデオストリームをオンに、VARIANT\_FALSE はオフにします。

#### 戻り値

値	意味
S_OK	メソッドは成功
NP_E_NOT_STARTED	カメラが起動していない
NP_E_DEVICE_DISCONNECTED	デバイスがシステムから取り除かれている

NP\_E\_DEVICE\_NOT\_SUPPORTED

接続されているデバイスは、この API でサポートされない古いタイプのハードウェアである

#### 4.1.2.2.7 INPCamera::GetFrame

このメソッドは、カメラからの直近のフレームを返します。

```
HRESULT GetFrame(LONG lTimeout, INPCameraFrame **ppFrame);
```

##### パラメータ

*lTimeout*

[in] フレームを待機する時間です（ミリ秒単位）。カメラからフレームが届かない場合は、そのスレッドはブロックされます。スレッドがブロックされないようにするには、0 を指定します。INFINITE を指定するとスレッドは無期限に待機します。ブロックされたスレッドは、カメラが停止したときに通知を受け取ります。

*ppFrame*

[out, retval] INPCameraFrame インタフェースのポインタ

##### 戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ppFrame が NULL
NP_E_DEVICE_DISCONNECTED	デバイスがシステムから取り除かれている
NP_E_DEVICE_NOT_SUPPORTEDED	この API でサポートされない古いタイプのハードウェアである
HRESULT_FROM_WIN32(ERROR_TIMEOUT)	カメラからのフレームを待機中にタイムアウトが発生

#### 4.1.2.2.8 INPCamera::DrawFrame

このメソッドは、指定のウィンドウハンドルに、カメラのイメージを描画します。

```
HRESULT DrawFrame(INPCameraFrame *pFrame, LONG hwnd);
```

##### パラメータ

*pFrame*

[in] 描画するカメラフレームを指定します。カメラフレームは、INPCamera の GetFrame メソッドを呼び出すことによって取得できます。

*hwnd*

[in] イメージを描画するウィンドウのウィンドウハンドルです。イメージのサイズは、NP\_OPTION\_DRAW\_SCALE オプションを設定することによって変更できます。

#### 戻り値

値	意味
S_OK	メソッドは成功
E_INVALIDARG	無効な hwnd
NP_E_DEVICE_DISCONNECTED	デバイスがシステムから取り除かれている
NP_E_DEVICE_NOT_SUPPORTED	接続されているデバイスは、この API でサポートされない古いタイプのハードウェアである

#### 4.1.2.2.9 INPCamera::ResetTrackedObject

このメソッドは、シングルドットトラッキングエンジンをリセットします。現在トラッキングされているオブジェクトの重みが、他のオブジェクトと同等になります。

**HRESULT ResetTrackedObject();**

#### パラメータ

なし

#### 戻り値

値	意味
S_OK	メソッドは成功
NP_E_DEVICE_DISCONNECTED	デバイスがシステムから取り除かれている
NP_E_DEVICE_NOT_SUPPORTED	接続されているデバイスは、この API でサポートされない古いタイプのハードウェアである

#### 4.1.2.2.10 INPCamera::GetOption

このメソッドは、指定したカメラのオプションの値を返します。

**HRESULT GetOption(LONG lOption, VARIANT \*pVal);**

#### パラメータ

*lOption*

[in] 設定するオプションを指定します。オプションは NP\_OPTION の列挙子によって定義されます。

pVal/

[out, retval] 指定したオプションの値を返します。オプションの値と VARIANT のタイプとの対応については、以下の表を参照してください。

オプション	タイプ	意味
NP_OPTION_STATUS_GREEN_ON_TRACKING	VARIANT_BOOL	デフォルトでは、オブジェクトがトラッキングされている際に、OptiTrack API は状態 LED をオン/オフする。この動作が望ましくない場合は、このオプションを VARIANT_FALSE に設定。
NP_OPTION_TRACKED_OBJECT_COLOR	VT_I4	DrawFrame メソッドが呼び出される際、現在トラッキングされているオブジェクトは、この色で描画される。デフォルトの値は緑色。色の指定には、RGB マクロを使用。
NP_OPTION_UNTRACKED_OBJECTS_COLOR	VT_I4	DrawFrame メソッドが呼び出される際、トラッキングされていないオブジェクトはこの色で描画される。デフォルトの値は赤色。色の指定には、RGB マクロを使用。
NP_OPTION_OBJECT_COLOR_OPTION	VT_I4	DrawFrame メソッドが呼び出される際に、オブジェクトの色のオプションによって、さまざまなオブジェクトの描画方法を指定する。 NP_OBJECT_COLOR_OPTION の列挙子でオプションを指定。
NP_OPTION_DRAW_SCALE	VT_R8	描画スケールのオプションは、DrawFrame が呼び出される際にカメライメージに適用される乗数。有効な値は 0.1 から 5.0。
NP_OPTION_THRESHOLD	VT_I4	シーンのライトは、ビデオのしきい値レベルを変更することによってフィルタリングできる。しきい値より低い値では、低い光量のオブジェクトをキャプチャできる。高い値では、低い光量のオブジェクトをフィルタリングして除去する。有効な値は NP_THRESHOLD_MIN から NP_THRESHOLD_MAX まで。しきい値は書き込みのみで、デバイスには保存されない。API の起動やデバイスの挿入の際は、しきい値はデバイスのデフォルト値とみなされる。
NP_OPTION_OBJECT_MASS_WEIGHT	VT_R8	トラッキング対象のオブジェクトを決定する際のオブジェクトのサイズの重み。デフォルト値は 1.0。

NP_OPTION_OBJECT_RATIO_WEIGHT	VT_R8	トラッキング対象のオブジェクトを決定する際のオブジェクトのレシオの重み。デフォルト値は 1.0。
NP_OPTION_PROXIMITY_WEIGHT	VT_R8	トラッキング対象のオブジェクトを決定する際の、他のオブジェクトへの近接度の重み。デフォルト値は 1.0。
NP_OPTION_STATIC_COUNT_WEIGHT	VT_R8	トラッキングされるオブジェクトを決定する際のオブジェクトの不動性の重み。デフォルト値は 1.0。
NP_OPTION_SCREEN_CENTER_WEIGHT	VT_R8	トラッキング対象のオブジェクトを決定する際の、イメージャの中心への近接度の重み。デフォルト値は 1.0。
NP_OPTION_LAST_OBJECT_TRACKED_WEIGHT	VT_R8	トラッキング対象のオブジェクトを決定する際の、最後にトラッキングされたオブジェクトの重み。デフォルト値は 2.0。
NP_OPTION_OBJECT_MASS_MIN	VT_R8	トラッキングするオブジェクトとしてみなされる、最小のピクセル数。デフォルト値は 3。
NP_OPTION_OBJECT_MASS_MAX	VT_R8	トラッキング対象のオブジェクトとしてみなされる、最大ピクセル数。デフォルト値は 200。
NP_OPTION_OBJECT_MASS_IDEAL	VT_R8	トラッキング対象のオブジェクトとしての理想的なピクセル数。デフォルト値は 100。
NP_OPTION_OBJECT_MASS_OUT_OF_RANGE	VT_R8	値が MIN と MAX の範囲をはずれた場合にオブジェクトのサイズに割り当てるスコア。デフォルト値は 0。
NP_OPTION_OBJECT_RATIO_MIN	VT_R8	トラッキング対象のオブジェクトとしてのオブジェクトのレシオの最小値。比率は Width / Height。デフォルト値は 0.25。
NP_OPTION_OBJECT_RATIO_MAX	VT_R8	トラッキングするオブジェクトとしてのオブジェクトのレシオの最大値。比率は Width / Height。デフォルト値は 4.0。
NP_OPTION_OBJECT_RATIO_IDEAL	VT_R8	トラッキング対象のオブジェクトとしてのオブジェクトのレシオの理想値。比率は Width / Height。デフォルト値は 1.0。
NP_OPTION_OBJECT_RATIO_OUT_OF_RANGE	VT_R8	値が MIN と MAX の範囲をはずれた場合に比率にアサインするスコア。デフォルト

		値は 0。
NP_OPTION_PROXIMITY_MIN	VT_R8	トラッキング対象のオブジェクトとしての、他のオブジェクトからの距離の最小値。デフォルト値は 3。
NP_OPTION_PROXIMITY_MAX	VT_R8	トラッキング対象のオブジェクトとしての、他のオブジェクトからの距離の最大値。デフォルト値は 300。
NP_OPTION_PROXIMITY_IDEAL	VT_R8	トラッキング対象のオブジェクトとしての、他のオブジェクトからの距離の理想値。デフォルト値は 20。
NP_OPTION_PROXIMITY_OUT_OF_RANGE	VT_R8	値が MIN と MAX の範囲をはずれた場合に近接度のアサインするスコア。デフォルト値は 0。
NP_OPTION_STATIC_COUNT_MIN	VT_R8	トラッキング対象のオブジェクトとしての、オブジェクトが動かないフレームの最小数。デフォルト値は 0。
NP_OPTION_STATIC_COUNT_MAX	VT_R8	トラッキング対象のオブジェクトとしての、オブジェクトが動かないフレームの最大数。デフォルト値は 200。
NP_OPTION_STATIC_COUNT_IDEAL	VT_R8	トラッキング対象のオブジェクトとしての、オブジェクトが動かないフレーム数の理想値。デフォルト値は 0。
NP_OPTION_STATIC_COUNT_OUT_OF_RANGE	VT_R8	値が MIN と MAX の範囲をはずれた場合に静止カウントにアサインするスコア。デフォルト値は 0。
NP_OPTION_SCREEN_CENTER_MIN	VT_R8	トラッキング対象のオブジェクトとしての、イメージャの中心からの最小の距離。デフォルト値は 0。
NP_OPTION_SCREEN_CENTER_MAX	VT_R8	トラッキングするオブジェクトとしての、イメージャの中心からの最大の距離。デフォルト値は 256。
NP_OPTION_SCREEN_CENTER_IDEAL	VT_R8	トラッキング対象のオブジェクトとしての、イメージャの中心からの理想の距離。デフォルト値は 0。
NP_OPTION_SCREEN_CENTER_OUT_OF_RANGE	VT_R8	値が MIN と MAX の範囲をはずれた場合に、画面中心からの距離に割り当てるスコア。デフォルト値は 0。
NP_OPTION_LAST_OBJECT_MIN	VT_R8	最後にトラッキングされたオブジェクトとして考慮するためのフレームの最小数。デフォルト値は 0。
NP_OPTION_LAST_OBJECT_MAX	VT_R8	最後にトラッキングされたオブジェクト

		として考慮するためのフレームの最大数。デフォルト値は 20。
NP_OPTION_LAST_OBJECT_IDEAL	VT_R8	最後にトラッキングされたオブジェクトとして考慮するためのフレーム数の理想値。デフォルト値は 0。
NP_OPTION_LAST_OBJECT_OUT_OF_RANGE	VT_R8	値が MIN と MAX の範囲をはずれた場合に最後のオブジェクトに割り当てるためのスコア。デフォルト値は 0。
NP_OPTION_STATUS_LED_ON_START	VARIANT_BOOL	デフォルトでは、Start メソッドが呼び出される際に、OptiTrack API が状態 LED をオンにする。この動作が望ましくない場合には、Start メソッドを呼び出す前に、このオプションを VARIANT_FALSE に設定。
NP_OPTION_ILLUMINATION_LEDS_ON_START	VARIANT_BOOL	デフォルトでは、Start メソッドが呼び出される際に、OptiTrack API がイルミネーション LED をオンにする。この動作が望ましくない場合には、Start メソッドを呼び出す前に、このオプションを VARIANT_FALSE に設定。
NP_OPTION_CAMERA_ROTATION	VT_I4	カメラを、通常の直立姿勢以外の向きで取り付ける場合に設定。カメラを横向き、または上下逆に取り付けた場合はこのオプションを設定。カメラの回転を指定するためには、NP_CAMERA_ROTATION の列挙子を使用。これらの設定を適用するためには、変換したいオブジェクトごとに INPObject::Transform メソッドを呼び出す必要がある。
NP_OPTION_MIRROR_X	VARIANT_BOOL	X 平面での動きを反転したい場合に使用。X 平面での動きをミラーリングするためには、このオプションを VARIANT_TRUE に設定。  これらの設定を適用するためには、変換したいオブジェクトごとに INPObject::Transform メソッドを呼び出す必要がある。
NP_OPTION_MIRROR_Y	VARIANT_BOOL	Y 平面での動きを反転したい場合に使用。Y 平面での動きをミラーリングするためには、このオプションを VARIANT_TRUE に設定。これらの設定を適用するためには、変換したいオブジェクトごとに INPObject::Transform メソッドを呼び出す必要がある。

NP_OPTION_SEND_EMPTY_FRAMES	VARIANT_BOOL	デフォルトでは、OptiTrack API はデータを持たないカメラフレームをフィルタリングする。API がすべてのカメラフレームを送信するようにするためには、このオプションを VARIANT_TRUE に設定。
NP_OPTION_CAMERA_ID	VT_I4	カメラに ID 番号を割り当てる。カメラは PC に送るフレームにこの ID をスタンプする。  この機能は OptiTrack FLEX:C120 と V100 カメラでのみサポート。
NP_OPTION_FRAME_RATE	VT_I4	カメラのフレームレートを調整。すべてのパーセント値が使用可能なわけではなく、使用できない値が選ばれた場合には、自動的に、適合する最も近い値が使われる。低い値のパーセント値を設定すると、フレームレートがスローダウンし、カメラの露出が長くなり、明るいイメージが得られる。範囲は 3 から 100（最高フレームレートに対するパーセント値）。デフォルト値は 100。  C120 カメラに対する有効な設定は、3、6、10、13、16、20、33、40、50、66、100。  V100 カメラに対する有効な設定は、25、50、100。  この機能は OptiTrack FLEX:C120 と V100 カメラでのみサポート。
NP_OPTION_EXPOSURE	VT_I4	カメラの電子的露出制御を調節し、より明るいイメージや、より暗いイメージを得るための設定。  C120 カメラでの範囲は 0 から 399（値が小さいほど暗いイメージ）。デフォルト値は 150。  V100 カメラでの範囲は 0 から 479（値が小さいほど暗いイメージ）。デフォルト値は 55。  この機能は、OptiTrack FLEX:C120 および V100 カメラでのみサポート。
NP_OPTION_VIDEO_TYPE	VT_I4	カメラの IR イルミネーション LED の強度（明るさ）を制御。範囲は 1 から 12（値が小さいほど、IR イルミネーションの発光は抑制されます）。デフォルト値は 12（最大イルミネーションパワー）。

NP_OPTION_INTENSITY	VT_14	<p>V100 カメラのみ。15 の値を使用することによってストロボイルミネーションモードが可能。このモードでは、フレーム開始の瞬間のみ IR LED をフルパワーにする。ストロボモードを使用する際は、最良の結果を得るためには、短い露出時間（100 以下）を推奨。</p> <p>この機能は、OptiTrack FLEX:C120 および V100 カメラでのみサポート。</p>
NP_OPTION_FRAME_DECIMATION	VT_14	<p>カメラの IR イルミネーション LED の強度（明るさ）を制御。範囲は 1 から 12（値が小さいほど、IR イルミネーションの発光は抑制されます）。デフォルト値は 12（最大イルミネーションパワー）。</p> <p>V100 カメラのみ。15 の値を使用することによってストロボイルミネーションモードが可能。このモードでは、フレーム開始の瞬間のみ IR LED をフルパワーにする。ストロボモードを使用する際は、最良の結果を得るためには、短い露出時間（100 以下）を推奨。</p> <p>この機能は、OptiTrack FLEX:C120 および V100 カメラでのみサポート。</p> <p>カメラの自動フレーム廃棄（間引き）機能を制御。これによって、カメラは最速のシャッタースピードで露出するとともに、PC への伝送フレーム数を（N フレームごとに）減少することができる。範囲は 0 から 5（値が大きいほど、自動廃棄されるフレームが多い）デフォルト値は 0（フレーム廃棄なし）。0=フレーム廃棄なし、1=1 フレームおきに送出、2=4 フレームおきに送出、3=8 フレームおきに送出、4=16 フレームおきに送出、5=32 フレームおきに送出。</p> <p>この機能は、OptiTrack FLEX:C120 および V100 カメラでのみサポート。</p>
NP_OPTION_MINIMUM_SEGMENT_LENGTH	VT_14	<p>カメラが受け入れるオブジェクトのしきい値のスライスにおける、水平方向セグメント幅の最小値を設定。小さなイメージノイズを除去する際に便利。最小長さよりも短いセグメントは廃棄される。範囲は 0 から 1023（0.5 ピクセル単位）。デフォルト値は 0（検知された全ピクセルを許可）。</p>

NP_OPTION_MAXIMUM_SEGMENT_LENGTH	VT_14	<p>この機能は、OptiTrack FLEX:C120 および V100 カメラでのみサポート。</p> <p>カメラが受け入れるオブジェクトのしきい値のスライスにおける、水平方向セグメント幅の最大値を設定。大きなイメージノイズを除去する際に便利。最大長さよりも長いセグメントは廃棄される。範囲は 0 から 1023 (0.5 ピクセル単位)。デフォルト値は 1023 (検知した全ピクセルを許可)。</p> <p>この機能は、OptiTrack FLEX:C120 および V100 カメラでのみサポート。</p>
NP_OPTION_WINDOW_EXTENTS_X	VT_14	<p>カメライメージのクリッピングウィンドウの左端を設定。範囲は 74 から 427。デフォルト値は 74。</p> <p>この機能は、OptiTrack FLEX:C120 および V100 カメラでのみサポート。</p>
NP_OPTION_WINDOW_EXTENTS_X_END	VT_14	<p>カメライメージのクリッピングウィンドウの右端を設定。範囲は 74 から 427。デフォルト値は 427。</p> <p>この機能は、OptiTrack FLEX:C120 および V100 カメラでのみサポート。</p>
NP_OPTION_WINDOW_EXTENTS_Y	VT_14	<p>カメライメージのクリッピングウィンドウの上端を設定。範囲は 11 から 299。デフォルト値は 11。ビデオが動作するためには、イメージの合計垂直方向高さは結果的には 4 の倍数でなければならない。</p> <p>この機能は、OptiTrack FLEX:C120 および V100 カメラでのみサポート。</p>
NP_OPTION_WINDOW_EXTENTS_Y_END	VT_14	<p>カメライメージのクリッピングウィンドウの下端を設定。範囲は 11 から 299。デフォルト値は 299。</p> <p>この機能は、OptiTrack FLEX:C120 および V100 カメラでのみサポート。</p>
NP_OPTION_RESET_FRAME_COUNT	N/A	<p>カメラのフレーム ID カウンタ (フレームごとに増加) を 0 にリセット。読み取り専用。</p> <p>INPCameraFrame::get_Id() も参照。</p> <p>この機能は、OptiTrack FLEX:C120 および V100 カメラでのみサポート。</p>
NP_OPTION_NUMERIC_DISPLAY_ON	VT_14	<p>カメラの数値 LED をオンにして、数字を表示。範囲は 0 から 99。</p>

NP_OPTION_NUMERIC_DISPLAY_OFF	N/A	この機能は、OptiTrack FLEX:C120 および V100 カメラでのみサポート。
NP_OPTION_SEND_FRAME_MASK	VT_14	<p>カメラの数値 LED をオフに設定。</p> <p>この機能は、OptiTrack FLEX:C120 および V100 カメラでのみサポート。</p> <p>拡張フレーム伝送オプション。この設定により、無効（壊れた）かつ/または空の（オブジェクトが検知されない）フレームがコールバック通知をトリガーするかどうかを制御。デフォルト値は 0。</p> <p>渡される値はビット単位のフラグのセット。以下のフラグが利用可能。</p> <ul style="list-style-type: none"> <li>・無効な（壊れた）フレームに対するコールバックを有効にする（注：フレームは処理されない）。</li> </ul> <p>NP_FRAME_SENDINVALID ( 0x02)</p> <ul style="list-style-type: none"> <li>・空の（オブジェクトが検出されない）フレームに対するコールバックを有効にする。</li> </ul> <p>NP_FRAME_SENDEMPTY (0x01)</p>
NP_OPTION_TEXT_OVERLAY_OPTION	VT_14	<p>フレームイメージをレンダリングする際に、その上に現在のフレームに関する有益な情報をオーバーレイするかどうかを制御。デフォルト値は 0。</p> <p>渡される値はビット単位のフラグのセット。以下のフラグが利用可能。</p> <ul style="list-style-type: none"> <li>・ビデオモードとフレーム ID に関する情報のオーバーレイを有効に設定。</li> </ul> <p>NP_TEXT_OVERLAY_HEADER (0x01)</p> <ul style="list-style-type: none"> <li>・フレームで検出されたオブジェクトに関する情報のオーバーレイを有効に設定。</li> </ul> <p>NP_TEXT_OVERLAY_OBJECT (0x02)</p> <ul style="list-style-type: none"> <li>・フレームで検出されたオブジェクトの可視性を増大する情報のオーバーレイを有効に設定。</li> </ul> <p>NP_TEXT_OVERLAY_OBJECT_HIGHLIGHT (0x04)</p>
NP_OPTION_SCORING_ENABLED	VARIANT_BOOL	<p>デフォルト値は TRUE。</p> <p>組み込みのオブジェクトスコアリングとランキングのアルゴリズムを無効する際</p>

NP_OPTION_GRAYSCALE_DECIMATION	VT_14	<p>に使用。スコアリングを無効にすることで、カメラが大量のマーカをトラッキングする際の処理負荷を大幅に減少させることができる。API のスコアリングを無効にするためには、このオプションを VARIANT_FALSE に設定。</p>
NP_OPTION_OBJECT_CAP	VT_14	<p>カメラのグレースケールイメージダウンサンプリング（リサイズ）機能を制御。全グレースケールフレームを伝送するには USB バンド幅が十分でない場合に、この機能を利用。</p> <p>デフォルト値は 0（ダウンサンプリングなしの 640x480）。4=4 分の 1 サイズ（160x120）、2=2 分の 1 サイズ（320x240）。</p> <p>この機能は、OptiTrack V100 カメラでのみサポート。</p> <p>フレームで検出されるオブジェクトの最大数を制御。返されるオブジェクト数を小さくすることによって、パフォーマンスを向上させることができる。デフォルト値は 500。</p>

**戻り値**

値	意味
S_OK	メソッドは成功
E_INVALIDARG	無効な hwnd
NP_E_DEVICE_DISCONNECTED	デバイスがシステムから取り除かれている
NP_E_DEVICE_NOT_SUPPORTED	接続されているデバイスは、この API でサポートされない古いタイプのハードウェアである

**4.1.2.2.11 INPCamera::SetOption**

このメソッドは、指定したカメラオプションの値を設定します。オプションのリストについては、GetOption メソッドを参照してください。

```
HRESULT SetOption(LONG lOption, VARIANT Val);
```

**パラメータ**

*lOption*

[in] 設定するオプションを指定します。オプションは NP\_OPTION の列挙子によって定義されます。

*Val*

[in] オプションの値を保持する VARIANT です。変数の型はオプションによって異なります。

#### 戻り値

値	意味
S_OK	メソッドは成功
E_INVALIDARG	無効なオプションの値
NP_E_DEVICE_DISCONNECTED	デバイスがシステムから取り除かれている
NP_E_DEVICE_NOT_SUPPORTEDED	接続されているデバイスは、この API でサポートされない古いタイプのハードウェアである

#### 4.1.2.2.12 INPCamera::GetFrameImage

このメソッドは、指定のフレームのイメージをユーザが提供したバッファへレンダリングします。

```
HRESULT GetFrameImage(INPCameraFrame *pFrame, INT PixelWidth, INT PixelHeight, INT ByteSpan, INT BitsPerPixel, BYTE *Buffer);
```

#### パラメータ

*pFrame*

[in] 描画するカメラフレームを指定。カメラフレームは、INPCamera の GetFrame メソッドを呼び出すことによって取得できます。

*PixelWidth*

[in] 描画するイメージの幅をピクセル単位で指定します。

*PixelHeight*

[in] 描画されるイメージの高さをピクセル単位で指定します。

*ByteSpan*

[in] イメージの水平ラインの幅をバイト単位で指定します。（この値が 0 の場合、BitsPerPixel の値に基づいて自動的に計算されます。）

*BitsPerPixel*

[in] ピクセルあたりのビット数で、イメージの色深度を指定します。

値	意味
8	ピクセルあたり 8 ビット。8 ビット値を明度に使用して表示。

16	ピクセルあたり 16 ビット。フォーマットは RGB565。
24	ピクセルあたり 24 ビット。フォーマットは RGB。
32	ピクセルあたり 32 ビット。フォーマットは RGBA。

*Buffer*

[in] フレームを描画するバッファを指定します。バッファオーバーランとメモリ破損を防ぐために、バッファは PixelHeight × ByteSpan のサイズ、あるいは、PixelWidth × PixelHeight × (BitsPerPixel / 8) のサイズ程度に設定してください。

## 戻り値

値	意味
S_OK	メソッドは成功

## 4.1.3 INPVector

## 4.1.3.1 プロパティ

## 4.1.3.1.1 INPVector::get\_Yaw

直前のフレームから計算されたヨーの量を返します。

```
HRESULT get_Yaw(VARIANT *pVal);
```

## パラメータ

*pVal*

[out, retval] ヨーを受け取る VARIANT 型のポインタです。

## 戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

## 4.1.3.1.2 INPVector::get\_Pitch

直前のフレームから計算された、ピッチの量を返します。

```
HRESULT get_Pitch(VARIANT *pVal);
```

## パラメータ

*pVal*

[out, retval] ピッチを受け取る VARIANT 型のポインタです。

戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

#### 4.1.3.1.3 INPVector::get\_Roll

直前のフレームから計算されたロールの量を返します。

```
HRESULT get_Roll(VARIANT *pVal);
```

パラメータ

*pVal*

[out, retval] ロールを受け取る VARIANT 型のポインタです。

戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

#### 4.1.3.1.4 INPVector::get\_X

前のフレームから計算された X の位置を返します。

```
HRESULT get_Yaw(VARIANT *pVal);
```

パラメータ

*pVal*

[out, retval] X の位置を受け取る VARIANT 型のポインタです。

戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

#### 4.1.3.1.5 INPVector::get\_Y

直前のフレームから計算された Y の位置を返します。

```
HRESULT get_Yaw(VARIANT *pVal);
```

パラメータ

*pVal*

[out, retval] Y の位置を受け取る VARIANT 型のポインタです。

**戻り値**

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

**4.1.3.1.6 INPVector::get\_Z**

[out, retval] Z の位置を受け取る VARIANT 型のポインタです。

```
HRESULT get_Z(VARIANT *pVal);
```

**パラメータ**

*pVal*

[out, retval] Z の位置を受け取る VARIANT 型のポインタです。

**戻り値**

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

**4.1.3.2 メソッド****4.1.3.2.1 INPVector::Update**

指定のカメラフレームに基づいて現在のベクトルの位置を計算するために、Update メソッドを呼び出します。Update の呼び出しが成功すると、INPVector インタフェースのプロパティは新しい値を保持します。

```
HRESULT Update(INPCamera *pCamera, INPCameraFrame *pFrame);
```

**パラメータ**

*pCamera*

[in] フレームを送出したカメラへのポインタです。

*pFrame*

[in] 値を計算するために使用したフレームへのポインタです。

**戻り値**

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

**4.1.3.2.2 INPVector::Reset**

Reset メソッドは、ベクトル計算を初期状態にリセットします。

```
HRESULT Reset();
```

**パラメータ**

なし

**戻り値**

値	意味
S_OK	メソッドは成功

#### 4.1.4 INPVector2

INPVector2 インタフェースは、INPVector が提供する機能の上に構築されています。INPVector2 は INPVector に属し、新しい機能に加えて、INPVector のすべてのメソッドをサポートします。

INPVector2 によって、Vector クリップの構成を変更することができます。カスタム Vector クリップは、2 つの点が第 3 の点よりも低い位置にある三角形でなくてはなりません。また、上方の点は下の 2 点からオフセットされている必要があります。このインタフェースによって、より大きな、あるいはより小さな三角形を作成することができます（カスタムシェイプは作成できません）。構成の変更には、4 つの値が関係します。dist01、dist02、dist12、dist0l です。下の図は、Vector クリップのレイアウトを示しています。



図 5 Vector レイアウト

dist01 - P0 と P1 の中心間の距離（ミリメートル単位）

dist02 - P0 と P2 の中心間の距離（ミリメートル単位）

dist12 - P1 と P2 の中心間の距離（ミリメートル単位）

dist0l - クリップを上から見た場合の、P0 から P1-P2 への垂線の距離

最適なパフォーマンスを得るには、dist01 と dist02 を等しくしてください。

下の図は、計測の様子を別のビューで示しています。

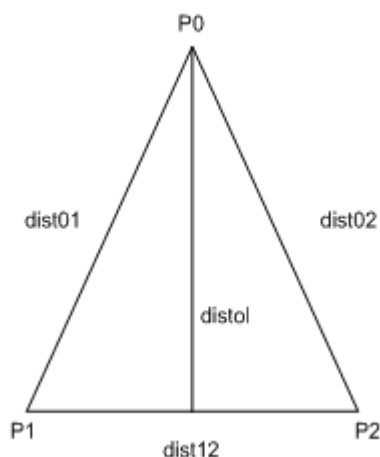


図6 見下ろした状態の Vector クリップ

#### 4.1.4.1 Properties

##### 4.1.4.1.1 INPVector2::get\_dist01

Vector クリップの P0 と P1 の間の距離（ミリメートル）を返します。

```
HRESULT get_dist01(VARIANT *pVal);
```

パラメータ

*pVal*

[out, retval] 値を受け取る VARIANT 型のポインタです。

戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

##### 4.1.4.1.2 INPVector2::put\_dist01

Vector クリップの P0 と P1 の間の距離を設定します（ミリメートル単位）。

```
HRESULT put_dist01(VARIANT Val);
```

パラメータ

*Val*

[in] 距離を指定する VARIANT です。VARIANT は VT\_R8 タイプでなくてはなりません。

戻り値

値	意味
S_OK	メソッドは成功
E_INVALIDARG	無効な値

#### 4.1.4.1.3 INPVector2::get\_dist02

Vector クリップの P0 と P2 の間の距離を返します（ミリメートル単位）。

```
HRESULT get_dist02(VARIANT *pVal);
```

パラメータ

*pVal*

[out, retval] 値を受け取る VARIANT 型のポインタです。

戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

#### 4.1.4.1.4 INPVector2::put\_dist02

Vector クリップの P0 と P2 の間の距離を設定します（ミリメートル単位）。

```
HRESULT put_dist02(VARIANT Val);
```

パラメータ

*Val*

[in] 距離を指定する VARIANT です。VARIANT は VT\_R8 タイプでなくてはなりません。

戻り値

値	意味
S_OK	メソッドは成功
E_INVALIDARG	無効な値

#### 4.1.4.1.5 INPVector2::get\_dist12

Vector クリップの P1 と P2 の間の距離を返します（ミリメートル単位）。

```
HRESULT get_dist12(VARIANT *pVal);
```

パラメータ

*pVal*

[out, retval] 値を受け取る VARIANT 型のポインタです。

戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

#### 4.1.4.1.6 INPVector2::put\_dist12

Vector クリップの P1 と P2 の間の距離を設定します（ミリメートル単位）。

```
HRESULT put_dist12(VARIANT Val);
```

パラメータ

*Val*

[in] 距離を指定する VARIANT です。VARIANT は VT\_R8 タイプでなくてはなりません。

戻り値

値	意味
S_OK	メソッドは成功
E_INVALIDARG	無効な値

#### 4.1.4.1.7 INPVector2::get\_dist01

Vector クリップの P0 から P1-P2 への垂線の距離を返します（ミリメートル単位）。

```
HRESULT get_dist01(VARIANT *pVal);
```

パラメータ

*pVal*

[out, retval] 値を受け取る VARIANT 型のポインタです。

戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

#### 4.1.4.1.8 INPVector2::put\_dist01

**注：この呼び出しは廃止されました。dist01 は、他の距離のパラメータを使って自動的に計算されます。**

Vector クリップの P0 から P1-P2 への垂線の距離を設定します（ミリメートル単位）。

```
HRESULT put_dist01(VARIANT Val);
```

パラメータ

*Val*

[in] 距離を指定する VARIANT です。VARIANT は VT\_R8 タイプでなくてはなりません。

戻り値

値	意味
S_OK	メソッドは成功
E_INVALIDARG	無効な値

#### 4.1.4.1.9 INPVector2::get\_Tracking

最後の更新の状態を返します。true の場合、ベクトルエンジンに渡された最後のフレームには3つの有効な点が含まれていたことを表します。

```
HRESULT get_Tracking(VARIANT *pVal);
```

##### パラメータ

*pVal*

[out, retval] 値を受け取る VARIANT 型のポインタです。

##### 戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

#### 4.1.4.2 メソッド

##### 4.1.4.2.1 INPVector2::GetPoint

GetPoint は Vector クリップの点のうちの1つの3D位置を返します。カメラフレームで Update メソッドを呼び出した後に、このメソッドを呼び出してください。データの有効性を保証するため、このメソッドを呼ぶ前にトラッキング状態をチェックしてください。

```
HRESULT GetPoint(int nPoint, INPPoint **ppPoint);
```

##### パラメータ

*nPoint*

[in] 問い合わせる点のインデックスです。有効な値の範囲は0から2。

*ppPoint*

[out, retval] 位置情報を受け取る INPPoint インタフェースへのポインタです。

##### 戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

#### 4.1.5 INPVector3

INPVector3 は、INPVector2 インタフェースに機能が追加されたものです。INPVector3 は INPVector に属し、新しい機能に加えて、INPVector と INPVector2 のすべてのメソッドをサポートします。

INPVector3 によって、ベクトル計算で使用するカメラの特性を変更することができます。最も重要なものとしては、ストックレンズ以外のレンズを利用する際に便利な、焦点距離の変更があります。

**4.1.5.1.1 INPVector2:: get\_imagerPixelWidth**

イメージの幅をピクセル単位で返します。

```
HRESULT get_imagerPixelWidth (VARIANT *pVal);
```

**パラメータ**

*pVal*

[out, retval] VT\_R8 タイプの値を受け取る VARIANT 型のポインタです。

**戻り値**

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

**4.1.5.1.2 INPVector2:: put\_imagerPixelWidth**

イメージの幅をピクセル単位で設定します。

```
HRESULT put_imagerPixelWidth (VARIANT Val);
```

**パラメータ**

*Val*

[in] 距離を指定する VARIANT です。VARIANT は VT\_R8 タイプでなくてはなりません。

**戻り値**

値	意味
S_OK	メソッドは成功
E_INVALIDARG	無効な値

**4.1.5.1.3 INPVector2:: get\_imagerPixelHeight**

イメージの高さをピクセル単位で返します。

```
HRESULT get_imagerPixelHeight (VARIANT *pVal);
```

**パラメータ**

*pVal*

[out, retval] VT\_R8 タイプの値を受け取る VARIANT 型のポインタです。

**戻り値**

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

#### 4.1.5.1.4 INPVector2:: put\_imagerPixelHeight

イメージの高さをピクセル単位で設定します。

```
HRESULT put_imagerPixelHeight (VARIANT Val);
```

パラメータ

*Val*

[in] 距離を指定する VARIANT です。VARIANT は VT\_R8 タイプでなくてはなりません。

戻り値

値	意味
S_OK	メソッドは成功
E_INVALIDARG	無効な値

#### 4.1.5.1.5 INPVector2:: get\_imagerMMWidth

イメージの幅をミリメートル単位で返します。

```
HRESULT get_imagerMMWidth (VARIANT *pVal);
```

パラメータ

*pVal*

[out, retval] VT\_R8 タイプの値を受け取る VARIANT 型のポインタです。

戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

#### 4.1.5.1.6 INPVector2:: put\_imagerMMWidth

イメージの幅をミリメートル単位で設定します。

```
HRESULT put_imagerMMWidth (VARIANT Val);
```

パラメータ

*Val*

[in] 距離を指定する VARIANT です。VARIANT は VT\_R8 タイプでなくてはなりません。

戻り値

値	意味
S_OK	メソッドは成功
E_INVALIDARG	無効な値

#### 4.1.5.1.7 INPVector2:: get\_imagerMMHeight

イメージの高さをミリメートル単位で返します。

```
HRESULT get_imagerMMHeight (VARIANT *pVal);
```

パラメータ

*pVal*

[out, retval] VT\_R8 タイプの値を受け取る VARIANT 型のポインタです。

戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

#### 4.1.5.1.8 INPVector2:: put\_imagerMMHeight

イメージの高さをミリメートル単位で設定します。

```
HRESULT put_imagerMMHeight (VARIANT Val);
```

パラメータ

*Val*

[in] 距離を指定する VARIANT です。VARIANT は VT\_R8 タイプでなくてはなりません。

戻り値

値	意味
S_OK	メソッドは成功
E_INVALIDARG	無効な値

#### 4.1.5.1.9 INPVector2:: get\_imagerMMFocalLength

イメージで使用されているレンズの焦点距離を、ミリメートル単位で返します。

TrackIR3/SmartNAV3/OptiTrack FLEX:3 stock lens = 3.7mm

TrackIR4 stock lens = 2.45mm

OptiTrack FLEX:G120 stock lens= 2.6mm

```
HRESULT get_imagerMMFocalLength (VARIANT *pVal);
```

パラメータ

*pVal*

[out, retval] VT\_R8 タイプの値を受け取る VARIANT 型のポインタです。

戻り値

値	意味
S_OK	メソッドは成功

E\_POINTER                      ポインタが無効

#### 4.1.5.1.10 INPVector2:: put\_imagerMMFocalLength

イメージャで使用するレンズの焦点距離をミリメートル単位で設定します。

```
HRESULT put_imagerMMFocalLength (VARIANT Val);
```

パラメータ

*Val*

[in] 距離を指定する VARIANT です。VARIANT は VT\_R8 タイプでなくてはなりません。

戻り値

値	意味
S_OK	メソッドは成功
E_INVALIDARG	無効な値

### 4.1.6 INPPoint

INPPoint インタフェースは、点の 3D 位置に関するラッパーです。3D の点の位置を返す必要がある、任意のインタフェースによって使用されます。

#### 4.1.6.1 プロパティ

##### 4.1.6.1.1 INPPoint::get\_X

任意の点の X 軸の位置を返します。

```
HRESULT get_X(VARIANT *pVal);
```

パラメータ

*pVal*

[out, retval] 位置を受け取る VARIANT 型のポインタです。戻される VARIANT は、VT\_R8 タイプです。

戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

##### 4.1.6.1.2 INPPoint::get\_Y

任意の点の Y 軸の位置を返します。

```
HRESULT get_Y(VARIANT *pVal);
```

パラメータ

*pVal*

[out, retval] 位置を受け取る VARIANT 型のポインタです。戻される VARIANT は、VT\_R8 タイプです。

#### 戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

#### 4.1.6.1.3 INPPoint::get\_Z

任意の点の Z 軸の位置を返します。

```
HRESULT get_Z(VARIANT *pVal);
```

#### パラメータ

*pVal*

[out, retval] 位置を受け取る VARIANT 型のポインタです。戻される VARIANT は、VT\_R8 タイプです。

#### 戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

### 4.1.7 INPSmoothing

#### 4.1.7.1 プロパティ

##### 4.1.7.1.1 INPSmoothing::get\_Amount

データに適用されているスムージングの量を返します。

```
HRESULT get_Smoothing(VARIANT *pVal);
```

#### パラメータ

*pVal*

[out, retval] スムージング量を受け取る VARIANT 型のポインタです。戻される VARIANT は、VT\_R8 タイプです。

#### 戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

#### 4.1.7.1.2 INPSmoothing::put\_Amount

データに適用されたスムージングの量を設定します。有効な値は、NP\_SMOOTHING\_MIN から NP\_SMOOTHING\_MAX です。

```
HRESULT put_Smoothing(VARIANT Val);
```

##### パラメータ

*Val*

[in] スムージング量を指定する VARIANT です。VARIANT は VT\_R8 タイプでなくてはなりません。

##### 戻り値

値	意味
S_OK	メソッドは成功
E_INVALIDARG	無効な量

#### 4.1.7.1.3 INPSmoothing::get\_X

スムージングした X の値を返します。

```
HRESULT get_X(VARIANT *pVal);
```

##### パラメータ

*pVal*

[out, retval] スムージングした X の値を受け取る VARIANT 型のポインタです。戻される VARIANT は、VT\_R8 タイプです。

##### 戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

#### 4.1.7.1.4 INPSmoothing::get\_Y

スムージングした Y の値を返します。

```
HRESULT get_Y(VARIANT *pVal);
```

##### パラメータ

*pVal*

[out, retval] スムージングした Y の値を受け取る VARIANT 型のポインタです。戻される VARIANT は、VT\_R8 タイプです。

##### 戻り値

値	意味
S_OK	メソッドは成功

E\_POINTER

ポインタが無効

## 4.1.7.2 メソッド

### 4.1.7.2.1 INPSmoothing::Update

新しい値に基づいてスムージングを計算するためには、Update メソッドを呼びます。Update の呼び出しが成功すると、INPSmoothing インタフェースのプロパティは新しい値を保持します。

```
HRESULT Update(VARIANT ValX, VARIANT ValY);
```

#### パラメータ

*ValX*

[in] 新しい X のデータを保持する VARIANT です。VARIANT は VT\_R8 タイプでなくてはなりません。

*ValY*

[in] 新しい Y のデータを保持する VARIANT です。VARIANT は VT\_R8 タイプでなくてはなりません。

#### 戻り値

値	意味
S_OK	メソッドは成功

### 4.1.7.2.2 INPSmoothing::Reset

Reset メソッドは、スムージング計算を初期状態にリセットします。

```
HRESULT Reset();
```

#### パラメータ

なし

#### 戻り値

値	意味
S_OK	メソッドは成功

## 4.1.8 INPCameraFrame

このオブジェクトは、現在のカメラフレームに関する情報を保持します。このインタフェースは、フレーム内のすべてのオブジェクトのリストを保持する、標準の COM の列挙型インタフェースです。

### 4.1.8.1 プロパティ

#### 4.1.8.1.1 INPCameraFrame::get\_\_NewEnum

このプロパティは列挙子のコピーを返します。返されるオブジェクトは、コレクションの最初のオブジェクトをポイントするインデックスを保持する、INPObject インタフェースです。

```
HRESULT get__NewEnum(LPUNKNOWN *ppunk);
```

パラメータ

*ppunk*

[out, retval] IUnknown インタフェースへのポインタです。

戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

#### 4.1.8.1.2 INPCameraFrame::get\_Count

コレクションのオブジェクトの数を返します。

```
HRESULT get_Count(LONG *pVal);
```

パラメータ

*pVal*

[out, retval] カウントを受け取る LONG 型のポインタです。

戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

#### 4.1.8.1.3 INPCameraFrame::get\_Id

このフレームに関連付けられた、ID 番号を返します。このメソッドは OptiTrack FLEX:C120 と V100 カメラでのみサポートされます。

```
HRESULT get_Id(LONG *pVal);
```

パラメータ

*pVal*

[out, retval] ID を受け取る LONG 型のポインタです。

戻り値

値	意味
---	----



E\_POINTER

ポインタが無効

**4.1.8.1.7 INPCameraFrame::get\_IsCorrupt**

このメソッドは、指定したフレームが壊れたデータを含むかどうかを表します。

壊れたフレームがコールバックをトリガーするかどうかを制御する NP\_OPTION\_SEND\_FRAME\_MASK も参照してください。

```
HRESULT get_IsCorrupt(VARIANT_BOOL *pVal);
```

**パラメータ**

*pVal*

[out, retval] 結果を受け取る LONG 型のポインタです。フレームが壊れたデータを含む場合、値は TRUE になります。

**戻り値**

値	意味
S_OK	メソッドは成功

**4.1.8.1.8 INPCameraFrame::get\_IsGreyscale**

このメソッドは、指定したフレームがグレースケールのイメージデータを含むかどうかを表します。

```
HRESULT get_IsGreyscale(VARIANT_BOOL *pVal);
```

**パラメータ**

*pVal*

[out, retval] 結果を受け取る LONG 型のポインタです。フレームがグレースケールのイメージデータを含む場合、値は TRUE になります。

**戻り値**

値	意味
S_OK	メソッドは成功

**4.1.8.1.9 INPCameraFrame::get\_IsEmpty**

このメソッドは、指定したフレームがオブジェクトを含むかどうかを表示します。

```
HRESULT get_IsEmpty(VARIANT_BOOL *pVal);
```

**パラメータ**

*pVal*

[out, retval] 結果を受け取る LONG 型のポインタです。フレームにオブジェクトが存在しない場合、値は TRUE になります。

**戻り値**

値	意味
---	----

S\_OK

メソッドは成功

## 4.1.8.2 メソッド

### 4.1.8.2.1 INPCameraFrame::Item

Item メソッドは、コレクションの中の対応するアイテムを返します。この例では、Item は INPObject インタフェースを返します。

```
HRESULT Item(LONG a_vlIndex, INPObject **ppObject);
```

#### パラメータ

*A\_vlIndex*

[in] 取得するアイテムのインデックスです。

*ppObject*

[out, retval] INPObject インタフェースを受け取るポインタです。

#### 戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

### 4.1.8.2.2 INPCameraFrame::Free

Free メソッドは、このフレームに対するすべての処理が完了した時点で呼び出す必要があります。システムにおいて、カメラフレームは限られたリソースであるため、可能な限り早期に解放しなければなりません。このメソッドが呼ばれた状態では、カメラフレームに依存する他のすべてのメソッドは、フレームへのアクセスの時点で失敗します。

```
HRESULT Free();
```

#### パラメータ

なし

#### 戻り値

値	意味
S_OK	メソッドは成功

## 4.1.9 INPObject

### 4.1.9.1 プロパティ

#### 4.1.9.1.1 INPObject::get\_Area

読み取り専用。オブジェクトの占める領域です（ピクセル単位）。返されるデータは DOUBLE 型です。

```
HRESULT get_Area(VARIANT *pVal);
```

**パラメータ**

*pVal*

[out, retval] オブジェクトの領域を受け取る VARIANT 型のポインタです。

**戻り値**

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

#### 4.1.9.1.2 INPObject::get\_X

読み取り専用。オブジェクトの X 位置です（ピクセル単位）。返されるデータは DOUBLE 型です。

```
HRESULT get_X(VARIANT *pVal);
```

**パラメータ**

*pVal*

[out, retval] オブジェクトの x 位置を受け取る VARIANT 型のポインタです。

**戻り値**

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

#### 4.1.9.1.3 INPObject::get\_Y

読み取り専用。オブジェクトの Y 位置です（ピクセル単位）。返されるデータは DOUBLE 型です。

```
HRESULT get_Y(VARIANT *pVal);
```

**パラメータ**

*pVal*

[out, retval] オブジェクトの Y 位置を受け取る VARIANT 型のポインタです。

**戻り値**

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

#### 4.1.9.1.4 INPObject::get\_Score

読み取り専用。オブジェクトの総スコアです。返されるデータは DOUBLE 型です。

```
HRESULT get_Score(VARIANT *pVal);
```

パラメータ

*pVal*

[out, retval] オブジェクトのスコアを受け取る VARIANT 型のポインタです。

戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

#### 4.1.9.1.5 INPObject::get\_Rank

読み取り専用。Rank は、カメラフレームにおけるオブジェクトのランクを特定します。ランクが1のオブジェクトが、トラッキング対象のオブジェクトです。

```
HRESULT get_Rank(LONG *pVal);
```

パラメータ

*pVal*

[out, retval] オブジェクトのランクを受け取る LONG 型のポインタです。

戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

#### 4.1.9.1.6 INPObject::get\_Width

読み取り専用。オブジェクト境界の矩形の幅を特定します。

```
HRESULT get_Rank(LONG *pVal);
```

パラメータ

*pVal*

[out, retval] 幅を受け取る LONG 型のポインタです。

戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

#### 4.1.9.1.7 INPObject::get\_Height

読み取り専用。オブジェクトの境界の矩形の高さを指定します。

```
HRESULT get_Rank(LONG *pVal);
```

##### パラメータ

*pVal*

[out, retval] 高さを受け取る LONG 型のポインタです。

##### 戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

#### 4.1.9.2 メソッド

##### 4.1.9.2.1 INPObject::Transform

所定のカメラに対して、指定されたオプションによる変換を適用します。変換には、カメラの回転と X 軸と Y 軸のミラーリングを含みます。このメソッドが呼び出されると、オブジェクトの X と Y の位置がわずかに変化します。通常、オブジェクト位置の原点は、デバイスの左上の角です。X の値は右方向に増加し、Y の値は下方向に増加します。

値を変換した後、xy 座標は、カメラの中心にある原点に対して相対的に計測されます。

```
HRESULT Transform(INPCamera *pCamera);
```

##### パラメータ

*pCamera*

[in] フレームを送出した INPCamera インタフェース型のポインタです。

##### 戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

#### 4.1.10 INPAvi

INPAvi インタフェースによって、トラッキング情報のビットマップイメージを AVI ファイルに書き込むことができます。AVI ファイルに使われる圧縮方式は、Microsoft MPEG4 v2 codec です。

AVI ファイルに保存されたイメージは、INPCamera::DrawFrame を呼び出すときに表示されるイメージと同一です。ビューにおける各オブジェクトの色は、標準的な NP\_OPTION の列挙子の値によって変更できます。

注： デフォルトでは、システムはクライアントアプリケーションに空のフレームを渡しません。これらのフレームを受け取るためには、NP\_OPTION\_SEND\_EMPTY\_FRAMES オプションを設定します。このオプションを設定しないと、可視のオブジェクトが存在するフレームのみが AVI ファイルに書き込まれます。

#### 4.1.10.1 プロパティ

##### 4.1.10.1.1 INPAvi::get\_FileName

AVI ファイルを保存する際に使うファイル名です。デフォルトのファイル名は optitrack.avi です。フルパスが指定されない場合は、ファイルはプロセスの現在の作業ディレクトリに置かれます。

```
HRESULT get_FileName(BSTR *pVal);
```

###### パラメータ

*pVal*

[out, retval] ファイル名を受け取る BSTR へのポインタです。

###### 戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

##### 4.1.10.1.2 INPAvi::put\_FileName

出力 AVI のファイル名を設定します。

```
HRESULT put_FileName(BSTR Val);
```

###### パラメータ

*Val*

[in] ファイル名に対する BSTR を保持する VARIANT です。

###### 戻り値

値	意味
S_OK	メソッドは成功

##### 4.1.10.1.3 INPAvi::get\_FrameRate

AVI ファイルに書き込まれる、1 秒あたりのフレーム数です。デフォルトでは 120 です。

```
HRESULT get_FileName(LONG *pVal);
```

###### パラメータ

*pVal*

[out, retval] フレームレートを受け取る LONG 型のポインタです。

#### 戻り値

値	意味
S_OK	メソッドは成功
E_POINTER	ポインタが無効

#### 4.1.10.1.4 INPAvi::put\_FrameRate

AVI ファイルのフレームレートを設定します。

```
HRESULT put_FileName(LONG Val);
```

#### パラメータ

*Val*

[in] フレームレートを保持する LONG です。

#### 戻り値

値	意味
S_OK	メソッドは成功

### 4.1.10.2 メソッド

#### 4.1.10.2.1 INPAvi::Start

書き込む AVI ファイルを初期化します。この関数は、AddFrame を呼び出す前に呼び出さなくてはなりません。

```
HRESULT Start();
```

#### パラメータ

なし

#### 戻り値

値	意味
S_OK	メソッドは成功

#### 4.1.10.2.2 INPAvi::Stop

完了の際に、AVI ファイルをクローズします。

```
HRESULT Stop();
```

#### パラメータ

なし

#### 戻り値

値	意味
---	----

S\_OK

メソッドは成功

#### 4.1.10.2.3 INPAvi::AddFrame

AVI ファイルにフレームを書き込みます。

```
HRESULT AddFrame(INPCamera *pCamera, INPCameraFrame *pFrame)
```

##### パラメータ

*pCamera*

[in] フレームを送出した INPCamera オブジェクトへのポインタです。

*pFrame*

[in] INPCameraFrame オブジェクトへのポインタです。これが、AVI ファイルに書き込まれるフレームです。

##### 戻り値

値

意味

S\_OK

メソッドは成功

### 4.1.11 \_INPCameraCollectionEvents

#### 4.1.11.1 メソッド

##### 4.1.11.1.1 \_INPCameraCollectionEvents::DeviceRemoval

このメソッドは、デバイスがシステムから取り除かれているときに呼び出されます。

```
void DeviceRemoval(INPCamera *pCamera);
```

##### パラメータ

*pCamera*

[in] INPCamera オブジェクトへのポインタです。カメラオブジェクトは、システムから取り除かれたオブジェクトです。

##### 戻り値

なし

##### 4.1.11.1.2 \_INPCameraCollectionEvents::DeviceArrival

このメソッドは、デバイスがシステムに追加されたときに呼び出されます。

```
void DeviceArrival(INPCamera *pCamera);
```

##### パラメータ

*pCamera*

[in] INPCamera オブジェクトへのポインタです。カメラオブジェクトは、システムに追加されたオブジェクトです。

##### 戻り値

なし

## 4.1.12 \_INPCameraEvents

### 4.1.12.1 メソッド

#### 4.1.12.1.1 \_INPCameraEvents::FrameAvailable

このメソッドは、カメラがフレームをキャプチャした際に呼び出されます。

```
void FrameAvailable(INPCamera *pCamera);
```

##### パラメータ

*pCamera*

[in] INPCamera オブジェクトへのポインタです。これが、フレームをキャプチャしたカメラオブジェクトです。直近のフレームを取得するためには、INPCamera の GetFrame メソッドを呼び出します。

##### 戻り値

なし

#### 4.1.12.1.2 \_INPCameraEvents::SwitchChange

このメソッドは、スイッチの状態が変化したときに呼び出されます。

```
void SwitchChange(INPCamera *pCamera, LONG lNewSwitchState);
```

##### パラメータ

*pCamera*

[in] INPCamera オブジェクトへのポインタです。これが、フレームをキャプチャしたカメラオブジェクトです。直近のフレームを取得するためには、INPCamera の GetFrame メソッドを呼び出します。

*lNewSwitchState*

[in] スイッチの新しい状態です。スイッチの状態は NP\_SWITCH\_STATE 列挙子によって定義されます。

##### 戻り値

なし

## 5 サンプルコード

### 5.1 VBScript

SDK では、いくつかの VBScript のサンプルファイルが提供されています。これらのスクリプトでは、カメラを列挙し、カメラから情報を取得し、データを問い合わせる方法を示しています。以下のスクリプトが提供されています。

#### camInfoLED.vbs

このスクリプトは、システムに接続されているカメラを列挙してそれらの情報を表示する方法のサンプルです。また、カメラの除去と追加に関する情報を取得するために、コネクションポイントを接続する方法も示しています。最後に、デバイスの LED の制御方法も示しています。

#### camData.vbs

このスクリプトは、デバイスからカメラフレームデータを取得する方法のサンプルです。camInfoLED スクリプト上に作成されており、フレームが利用可能となった際に通知を取得します。

#### camSwitch.vbs

このスクリプトは、スイッチの状態に関する情報を取得する方法を示しています。スクリプトでは、通知を得るためにカメラオブジェクトのコネクションポイントを使用しています。

#### camVector.vbs

このスクリプトは、ベクトル位置を計算する方法のサンプルです。6 つのパラメータすべてを計算しています。

#### camRecord.vbs

このスクリプトは、トラッキングビューを AVI ファイルに保存する方法のサンプルです。

### 5.2 VB.NET 2003

SDK では、Visual Basic .NET 2003 で記述したサンプルアプリケーションが提供されています。このアプリケーションは一度に 1 つのカメラしか制御できませんが、システムに接続されているカメラを切り替えることができます。API を通じて利用可能なすべてのオプションには、メインダイアログとは別のオプションダイアログからアクセスできます。デバイス通知とデータ提供には、コネクションポイントのコールバックが使われています。

### 5.3 VC6

Visual C/C++ 6.0 プロジェクトでは、C/C++サンプルアプリケーションが入手できます。VC6 プロジェクトは、依然として VC6 を使っている開発者が多いことから、最大限の互換性を目的として選択されました。VC6 プロジェクトから VS.NET プロジェクトへの変換は簡単です。

このアプリケーションでは、OptiTrack API のすべてのコンポーネントを機能させます。カメラの列挙、データの収集、スムージング、ベクトル位置の計算を行うことができます。

また、カメライメージの描画も行います。サンプルには、コールバックのための接続ポイントを有効化するコードも含まれています。