***INuiFusionColorReconstruction接口***

<https://msdn.microsoft.com/en-us/library/microsoft.kinect.nuikinectfusioncolorvolume.inuifusioncolorreconstruction.aspx>

封装带有颜色的重建容积的创建，更新和捕捉函数。

interface INuiFusionColorReconstruction : public IUnknown

***INuiFusionMesh 接口***

<https://msdn.microsoft.com/en-us/library/microsoft.kinect.nuikinectfusionvolume.inuifusionmesh.aspx>

当捕捉重建容积时的捕捉项目。它提供捕捉的顶点、法线和三角索引。

interface INuiFusionMesh : public IUnknown

***INuiFusionMesh::GetNormals 方法***

<https://msdn.microsoft.com/en-us/library/microsoft.kinect.nuikinectfusionvolume.inuifusionmesh.getnormals.aspx>

得到捕捉法线。

public:

HRESULT GetNormals (

const Vector3 \*\*pNormals

)

pNormals 类型：vector3 捕捉法线。

返回值 类型：HRESULT 如果成功，返回S\_OK;否则返回错误代码。

***INuiFusionMesh::GetTriangleIndices 方法***

<https://msdn.microsoft.com/en-us/library/microsoft.kinect.nuikinectfusionvolume.inuifusionmesh.gettriangleindices.aspx>

得到捕捉的三角顶点。

public:

HRESULT GetTriangleIndices(

const int \*\*pTriangleVertexIndices

)

pTriangleVertexIndices 类型：int 捕捉的三角顶点。

返回值 类型：HRESULT 如果成功，返回S\_OK;否则返回错误代码。

***INuiFusionMesh::GetVertices 方法***

<https://msdn.microsoft.com/en-us/library/microsoft.kinect.nuikinectfusionvolume.inuifusionmesh.getvertices.aspx>

得到捕捉顶点。

public:

HRESULT GetVertices (

const Vector3 \*\*pVertices

)

pVertices 类型：Vector3 捕捉顶点。

返回值 类型：HRESULT 如果成功，返回S\_OK;否则返回错误代码。

***INuiFusionMesh::TriangleVertexIndexCount 方法***

<https://msdn.microsoft.com/en-us/library/microsoft.kinect.nuikinectfusionvolume.inuifusionmesh.trianglevertexindexcount.aspx>

得到捕捉的三角顶点索引数量。

public:

UINT TriangleVertexIndexCount (

void

)

返回值 类型：UINT 捕捉的三角顶点索引数量。

***INuiFusionMesh::VertexCount 方法***

<https://msdn.microsoft.com/en-us/library/microsoft.kinect.nuikinectfusionvolume.inuifusionmesh.vertexcount.aspx>

得到捕捉里的顶点数量。

public:

UINT VertexCount (

void

)

返回值 类型：UINT 捕捉里的顶点数量。

***INuiFusionReconstruction接口***

<https://msdn.microsoft.com/en-us/library/microsoft.kinect.nuikinectfusionvolume.inuifusionreconstruction.aspx>

封装重建容积的创建更新和捕捉函数。

interface INuiFusionReconstruction : public IUnknown

***INuiFusionReconstruction::CalculateMesh 方法***

<https://msdn.microsoft.com/en-us/library/microsoft.kinect.nuikinectfusionvolume.inuifusionreconstruction.calculatemesh.aspx>

从带有每顶点颜色的重建容积输出一个零交点稠密表面的模型捕捉。

public:

HRESULT CalculateMesh(

UINT voxelStep,

INuiFusionMesh \*\*ppMesh

)

voxelStep 类型: UINT 当输出捕捉时在容积中取样体素点的步长值，它决定了捕捉的分辨率。使用大值生成低分辨率的捕捉。 这个值必须大于0，小于最小的轴体素分辨率。要使用满分辨率捕捉容积，使用步长值为1。这个参数中任何大于1的值都会有丢失零点的风险，并且因此丢失表面或表面细节。

ppMesh 类型: INuiFusionMesh 生成的捕捉项目。

返回值 类型：HRESULT 成功返回S\_OK；否则返回失败代码。

***INuiFusionReconstruction::CalculatePointCloud方法***

<http://msdn.microsoft.com/enus/library/microsoft.kinect.nuikinectfusionvolume.inuifusionreconstruction.calculatepointcloud.aspx>

通过在重建容积内进行光线投射计算点云，返回点云包括三维点和 来自指定相机姿势和颜色可视化图像的可视像素的零点稠密表面的法线。

public:

HRESULT CalculatePointCloud(

const NUI\_FUSION\_IMAGE\_FRAME \*pPointCloudFrame,

const Matrix4 \*pWorldToCameraTransform

)

pPointCloudFrame 类型: NUI\_FUSION\_IMAGE\_FRAME 预先分配的点云帧，填充由光线投射的重建容积。这个点云能用来当作一个在接下来调用AlignPointClouds方法的参考帧，或者传递到NuiFusionShadePointCloud函数中来产生一个可视化图像输出。

点云帧可以是一个任意的图像大小。这允许你以你的窗口大小计算点云，然后通过调用NuiFusionShadePointCloud方法创建一个可视化图像来呈现这个图像。记住大图的计算是昂贵的。

pWorldToCameraTransform 类型: Matrix4 光线投射的世界到相机转换（相机姿势）。

Return value 类型: HRESULT S\_OK if successful; otherwise, returns a failure code.

***INuiFusionReconstruction::DepthToDepthFloatFrame方法***

<http://msdn.microsoft.com/enus/library/microsoft.kinect.nuikinectfusionvolume.inuifusionreconstruction.depthtodepthfloatframe.aspx>

将指定的Kinect深度像素数组(以毫米为单位无符号整型)转化为一个NUI\_FUSION\_IMAGE\_FRAME结构（以米为单位浮点型）。

public:

HRESULT DepthToDepthFloatFrame(

const NUI\_DEPTH\_IMAGE\_PIXEL \*pDepthImageData,

UINT countDepthImageDataBytes,

const NUI\_FUSION\_IMAGE\_FRAME \*pDepthFloatFrame,

FLOAT minDepthClip,

FLOAT maxDepthClip,

BOOL mirrorDepth

)

pDepthImageData 类型: NUI\_DEPTH\_IMAGE\_PIXEL 无符号短整型Kinect深度像素数组的指针来转换。这个数据必须与作为pDepthFloatFrame参数的结构指针有相同的像素分辨率。

countDepthImageDataBytes 类型: UINT pDepthImageData参数指向数据的大小，以字节为单位。

pDepthFloatFrame 类型: NUI\_FUSION\_IMAGE\_FRAME 一个浮点深度帧的指针，来接收转换的深度数据。这个数据必须与作为pDepthImageData参数的数据指针有相同的像素分辨率。

minDepthClip 类型: FLOAT 最小深度的阈值，以米为单位。小于这个阈值的值将被设置为0.

maxDepthClip 类型: FLOAT 最大的深度阈值，以米为单位。大于这个阈值的值将被设置成1000.

mirrorDepth 类型: BOOL 镜像反映深度值，指定正确。如果从后面的kinect传感器来看图像出现了纠正，指定错误。

Return Value 类型: HRESULT HRESULT success or error code.

minDepthClip和maxDepthClip 参数使能裁剪输入数据;例如，这些值可以帮助隔离特定的对象或表面来重建。 在相机是静止的或移动不明显的情况下，minDepthCilp参数对于集成重建是重要的，它使任何沿着射线靠近相机的体素被剔除掉而不是保留(如果设置为0，忽略处理)。当重建大的真实世界大小的容积，当相机圆周移动，maxDepthClip参数是重要的，在视线中远离传感器比这个阈值大的体素将被移除。

这个方法在GPU中运行。

***INuiFusionReconstruction::GetCurrentWorldToCameraTransform方法***

<http://msdn.microsoft.com/enus/library/microsoft.kinect.nuikinectfusionvolume.inuifusionreconstruction.getcurrentworldtocameratransform.aspx>

得到现在内部的世界到相机转换（相机视图构成）。

public:

HRESULT GetCurrentWorldToCameraTransform(

Matrix4 \*pWorldToCameraTransform

)

pWorldToCameraTransform 类型: Matrix4 一个预先分配的Matrix4指针，填充现在的相机姿势

Return Value 类型: HRESULT Returns S\_OK if successful; otherwise, returns one of the following error codes:

E\_POINTER pWorldToCameraTransform参数为空。

E\_FAIL 未知原因的调用失败。

***INuiFusionReconstruction::GetCurrentWorldToVolumeTransform方法***

<http://msdn.microsoft.com/enus/library/microsoft.kinect.nuikinectfusionvolume.inuifusionreconstruction.getcurrentworldtovolumetransform.aspx>

得到现在内部的世界到容积转换。

public:

HRESULT GetCurrentWorldToVolumeTransform(

Matrix4 \*pWorldToVolumeTransform

)

pWorldToVolumeTransform 类型: Matrix4 一个预先分配的Matrix4指针，填充世界坐标到重建容积坐标的相似转换。这个转换表示世界坐标体系的起源在容积坐标和世界坐标到容积索引转换的位置和方向。

Note:使用一个正确的传递容积坐标，在立方体的前平面的左上角要有容积原点(即体素0，0，0)。类似于位图左上角原点，+x向左，+y向上，+z向前从原点到重建容积。默认的转换是在x，y在重建容积正面的中心确定世界的原点，并且缩放重建参数的体素每米来将世界坐标转化为容积体素。

***INuiFusionReconstruction::ProcessFrame方法***

<http://msdn.microsoft.com/enus/library/microsoft.kinect.nuikinectfusionvolume.inuifusionreconstruction.processframe.aspx>

一个高层函数通过KinectFusion管道来处理深度帧。

特别地，它对每一帧在设备上对等地处理相当于以下函数：1.AlignDepthFloatToReconstruction 2.IntergrateFrame

用户也可以选择单独调用底层函数替代这个函数，来得到更灵活的控制。然而，由于集成自然调用，这个函数将更快。这个函数调用成功后，如果需要一个输出的重建图像，用户可以继续调用CalculatePointCloud和ShadePointCloud函数。

这个函数最大支持分辨率640x480.

Note:如果在AlignDepthFloatToReconstruction阶段有追踪错误，将没有深度数据集成生成，内部相机姿势也保持不变.。

public:

HRESULT ProcessFrame(

const NUI\_FUSION\_IMAGE\_FRAME \*pDepthFloatFrame,

USHORT maxAlignIterationCount,

USHORT maxIntegrationWeight,

const Matrix4 \*pWorldToCameraTransform

)

pDepthFloatFrame 类型: NUI\_FUSION\_IMAGE\_FRAME 一个来自Kinect相机的浮点深度帧指针。

maxAlignIterationCount 类型: USHORT AlignDepthFloatToReconstruction运行的最大迭代数。最小值是1。 只使用小数目的迭代转换将运行更快，但是这算法将不会收敛到正确的转换。

maxIntegrationWeight 类型: USHORT 一个来控制当时深度迭代的平滑度的参数。最小值是1。小值有更多的噪点，但是对象能够集成和分解得更快，所以更适合动态环境。大值集成对象慢，但是能够提供更好的细节和降低噪点。

pWorldToCameraTransform 类型: Matrix4

可选的，一个预先分配的Matrix4指针，由最近的相机姿势猜测构成（例如上一个过程调用得到的相机姿势）。 如果传递的是NULL，则由最近的AlignDepthFloatToReconstruction,ProxessFrame或者IntergrateFrame调用得到内部存储姿势来初始化追踪。

Return Value 类型: HRESULT Returns S\_OK if successful; otherwise, returns one of the following failure codes:

E\_INVALIDARG The pDepthFloatFrame parameter is NULL, or the pDepthFloatFrame parameter is an incorrect image size or type, or the maxAlignIterationCount parameter or the maxIntegrationWeight parameter is less than 1.

E\_NOINTERFACE The pDepthFloatFrame parameter's pFrameTexture member is NULL.

E\_NUI\_FUSION\_TRACKING\_ERROR AlignDepthFloatToReconstruction 算法失败或没有融合

E\_NUI\_FEATURE\_NOT\_INITIALIZED The Kinect camera runtime could not be accessed.

E\_NUI\_DEVICE\_NOT\_CONNECTED No Kinect camera could be found.

E\_FAIL The call failed for an unknown reason.

***INuiFusionReconstruction::ResetReconstruction方法***

<http://msdn.microsoft.com/enus/library/microsoft.kinect.nuikinectfusionvolume.inuifusionreconstruction.resetreconstruction.aspx>

清除重建容积并且放一个新的世界到相机转换（相机视角姿势）。

public:

HRESULT ResetReconstruction(

const Matrix4 \*pInitialWorldToCameraTransform,

const Matrix4 \*pWorldToVolumeTransform

)

pInitialWorldToCameraTransform 类型: Matrix4 重建容积最初的相机姿态，带有世界原点的代表。要使用默认相机姿态，指定身份。

pWorldToVolumeTransform 类型: Matrix4

世界到容积转换表示世界坐标体系的原点在容积坐标和世界坐标到容积索引转换的位置和方向。这控制重建坐标体积出现在真实世界的哪代表世界的原始位置，或者代表相机如果身份在initialWorldToCameraTransform参数中通过的话。 (这将使世界起源和相机一致)。

Return value 类型: HRESULT S\_OK if successful; otherwise, returns a failure code.

要创建你自己的世界到容积转换，调用GetCurrentWorldToVolukeTransform方法，并且然后直接修改返回矩阵或者叠加你自己的相似矩阵，来修改容积转换或旋转代表世界坐标的体系。

要重置容积的同时保持相同的世界到容积转换，调用GetCurrentWorldToVolumeTransform方法，并且然后将返回的矩阵放到ResetReconstruction方法的pWorldToVolumeTransform参数上。

***NuiFusionCreateImageFrame函数***

<http://msdn.microsoft.com/enus/library/microsoft.kinect.nuikinectfusiondepthprocessor.nuifusioncreateimageframe.aspx>

创建一个新的NUI\_FUSION\_IMAGE\_FRAME对象。

HRESULT FUSIONAPI NuiFusionCreateImageFrame(

NUI\_FUSION\_IMAGE\_TYPE frameType,

UINT width,

UINT height,

const NUI\_FUSION\_CAMERA\_PARAMETERS \*pCameraParameters,

NUI\_FUSION\_IMAGE\_FRAME \*\*ppImageFrame

)

frameType 类型: NUI\_FUSION\_IMAGE\_TYPE [in] 创建的图像帧的类型。

width 类型: UINT [in] 帧的宽度，以像素为单位

height 类型: UINT [in] 帧的宽度，以像素为单位

pCameraParameters 类型: NUI\_FUSION\_CAMERA\_PARAMETERS [in, optional] 与帧相关联的相机形参

ppImageFrame 类型: NUI\_FUSION\_IMAGE\_FRAME [out] 新的 NUI\_FUSION\_IMAGE\_FRAME对象指针。

Return value 类型: HRESULT FUSIONAPI S\_OK if successful; otherwise, returns a failure code.

***NuiFusionCreateReconstruction函数***

<http://msdn.microsoft.com/enus/library/microsoft.kinect.nuikinectfusionvolume.nuifusioncreatereconstruction.aspx>

初始化一个新的INuiFusionReconstruction类。

HRESULT FUSIONAPI NuiFusionCreateReconstruction(

const NUI\_FUSION\_RECONSTRUCTION\_PARAMETERS \*pReconstructionParameters, NUI\_FUSION\_RECONSTRUCTION\_PROCESSOR\_TYPE reconstructionProcessorType,

INT deviceIndex,

const Matrix4 \*pInitialWorldToCameraTransform,

INuiFusionReconstruction \*\*ppNuiFusionReconstruction )

pReconstructionParameters 类型：NUI\_FUSION\_RECONSTRUCTION\_PARAMETERS [in] 容积重建参数。

reconstructionProcessorType 类型: NUI\_FUSION\_RECONSTRUCTION\_PROCESSOR\_TYPE [in]重建处理使用的模式（APU还是GPU）。

deviceIndex 类型: INT [in] 重建使用设备的索引。

pInitialWorldToCameraTransform 类型: Matrix4 [in, optional] 最初的世界到相机转换（坐标转换矩阵）。

ppNuiFusionReconstruction 类型: INuiFusionReconstruction [out] 返回的新的INuiFusionReconstruction接口的指针。

类型: HRESULT FUSIONAPI

S\_OK 成功; 否则返回错误代码。

***NuiFusionGetDeviceInfo函数***

<http://msdn.microsoft.com/enus/library/microsoft.kinect.nuikinectfusionvolume.nuifusiongetdeviceinfo.aspx>

提供关于Kinect Fusion重建处理设备的信息。

HRESULT FUSIONAPI NuiFusionGetDeviceInfo(

\_NUI\_FUSION\_RECONSTRUCTION\_PROCESSOR\_TYPE type,

INT index,

PWSTR pDescription,

UINT descriptionSizeInChars,

PWSTR pInstancePath,

UINT instancePathSizeInChars,

UINT \*pMemoryKB

)

type 类型: \_NUI\_FUSION\_RECONSTRUCTION\_PROCESSOR\_TYPE [in] 设备的重建处理模式。

index 类型: INT [in] 设备的索引

pDescription 类型: PWSTR 设备的描述。

descriptionSizeInChars 类型: UINT [in] 描述的大小，以位为单位

pInstancePath 类型: PWSTR 用来重建的GPU的DirectX实例路径

instancePathSizeInChars 类型: UINT [in]实例路径的大小，以位为单位

pMemoryKB 类型: UINT [out, optional] 得到在GPU上用来重建的专用显存数量。

Return value 类型: HRESULT FUSIONAPI S\_OK if successful; otherwise, returns a failure code.

***NuiFusionShadePointCloud函数***

<http://msdn.microsoft.com/enus/library/microsoft.kinect.nuikinectfusiondepthprocessor.nuifusionshadepointcloud.aspx>

通过点云帧产生两个灰度图，基于点的位置和表面标准。

HRESULT FUSIONAPI NuiFusionShadePointCloud(

const NUI\_FUSION\_IMAGE\_FRAME \*pPointCloudFrame,

const Matrix4 \*pWorldToCameraTransform,

const Matrix4 \*pWorldToBGRTransform,

const NUI\_FUSION\_IMAGE\_FRAME \*pShadedSurfaceFrame,

const NUI\_FUSION\_IMAGE\_FRAME \*pShadedSurfaceNormalsFrame

)

pPointCloudFrame 类型: NUI\_FUSION\_IMAGE\_FRAME [in] 要灰度化的点云帧。

pWorldToCameraTransform 类型：Matrix4 [in] 用来确定与世界有关的灰度点云透视的转换。这会影响应用worldToRGBTransform到点云。为达到一致的结果，它应该匹配用于建立点云帧的世界到相机转换

pWorldToBGRTransform 类型: Matrix4 [in, optional]这转换定义了一个从位置到RGB颜色空间的映射。矩阵转换来的点的x、y、z在渲染帧里被用作r、g、b的值。

pShadedSurfaceFrame 类型: NUI\_FUSION\_IMAGE\_FRAME [in, optional] 一个颜色图像帧，用来接收包含基于表面法线的光照的灰度帧。

pShadedSurfaceNormalsFrame 类型: NUI\_FUSION\_IMAGE\_FRAME [in, optional]一个颜色图像帧，用来接收灰度帧。

Return value 类型: HRESULT FUSIONAPI S\_OK if successful; otherwise, returns a failure code.

***NUI\_DEPTH\_IMAGE\_PIXEL结构***

<http://msdn.microsoft.com/en-us/library/nuiimagecamera.nui_depth_image_pixel.aspx>

包含一个深度数据像素。

typedef struct \_NUI\_DEPTH\_IMAGE\_PIXEL {

USHORT playerIndex;

USHORT depth;

} NUI\_DEPTH\_IMAGE\_PIXEL;

playerIndex 该像素的用户索引

depth 像素的深度

***NUI\_FUSION\_CAMERA\_PARAMETERS结构***

<http://msdn.microsoft.com/enus/library/microsoft.kinect.nuikinectfusioncolorvolume.nui_fusion_camera_parameters.aspx>

储存内在相机参数。这些参数描述了相机镜头的光学系统和传感器。

typedef struct \_NUI\_FUSION\_CAMERA\_PARAMETERS {

FLOAT focalLengthX;

FLOAT focalLengthY;

FLOAT principalPointX;

FLOAT principalPointY;

} NUI\_FUSION\_CAMERA\_PARAMETERS;

focalLengthX

The focal length for X normalized by the camera width.

focalLengthY

The focal length for Y normalized by the camera height.

principalPointX

The principal point for X normalized by the camera width.

principalPointY

The principal point for Y normalized by the camera height.

***NUI\_FUSION\_CAMERA\_POSE\_FINDER\_PARAMETERS结构***

<http://msdn.microsoft.com/enus/library/microsoft.kinect.nuikinectfusioncameraposefinder.nui_fusion_camera_pose_finder_parameters.aspx>

***NUI\_FUSION\_IMAGE\_FRAME结构***

<http://msdn.microsoft.com/enus/library/microsoft.kinect.nuikinectfusionvolume.nui_fusion_image_frame.aspx>

图像帧参数。

typedef struct \_NUI\_FUSION\_IMAGE\_FRAME {

UINT width;

UINT height;

NUI\_FUSION\_IMAGE\_TYPE imageType;

NUI\_FUSION\_CAMERA\_PARAMETERS \*pCameraParameters;

INuiFrameTexture \*pFrameTexture;

} NUI\_FUSION\_IMAGE\_FRAME;

width 创建图像的宽度

height 创建图像的高度

imageType 创建图像的类型

pCameraParameters 图像的相机参数

pFrameTexture 原始纹理的指针

***NUI\_FUSION\_RECONSTRUCTION\_PARAMETERS结构***

<http://msdn.microsoft.com/enus/library/microsoft.kinect.nuikinectfusioncolorvolume.nui_fusion_reconstruction_parameters.aspx>

包含Kinect Fusion重建参数。

typedef struct \_NUI\_FUSION\_RECONSTRUCTION\_PARAMETERS {

FLOAT voxelsPerMeter;

UINT voxelCountX;

UINT voxelCountY;

UINT voxelCountZ;

} NUI\_FUSION\_RECONSTRUCTION\_PARAMETERS;

voxelsPerMeter 每米体积像素数量

voxelCountX 重建X轴体积像素数量

voxelCountY 重建Y轴体积像素数量

voxelCountZ 重建Z轴体积像素数量

SDK中使用的数据为:

m\_reconstructionParams.voxelsPerMeter = 256.f;

m\_reconstructionParams.voxelCountX = 384;

m\_reconstructionParams.voxelCountY = 384;

m\_reconstructionParams.voxelCountZ = 384;

即每米有256个体素，每两个临近的体素距离约为4mm.

建立了1.5m\*1.5m\*1.5m的容积，消耗内存384\*384\*384\*4，200+MB(可能还有内存对齐消耗的内存)

***NUI\_FUSION\_IMAGE\_TYPE枚举***

<http://msdn.microsoft.com/enus/library/microsoft.kinect.nuikinectfusionvolume.nui_fusion_image_type.aspx>

typedef enum \_NUI\_FUSION\_IMAGE\_TYPE

{

NUI\_FUSION\_IMAGE\_TYPE\_INVALID = 0,

NUI\_FUSION\_IMAGE\_TYPE\_COLOR = 1,

NUI\_FUSION\_IMAGE\_TYPE\_FLOAT = 2,

NUI\_FUSION\_IMAGE\_TYPE\_POINT\_CLOUD = 3

} NUI\_FUSION\_IMAGE\_TYPE;

NUI\_FUSION\_IMAGE\_TYPE\_INVALID 无效的图像

NUI\_FUSION\_IMAGE\_TYPE\_COLOR 颜色图像

NUI\_FUSION\_IMAGE\_TYPE\_FLOAT 浮动图像

NUI\_FUSION\_IMAGE\_TYPE\_POINT\_CLOUD 点云图像

***NUI\_FUSION\_RECONSTRUCTION\_PROCESSOR\_TYPE枚举***

<http://msdn.microsoft.com/enus/library/microsoft.kinect.nuikinectfusionvolume.nui_fusion_reconstruction_processor_type.aspx>

指定重建处理模式。

enum \_NUI\_FUSION\_RECONSTRUCTION\_PROCESSOR\_TYPE

{

NUI\_FUSION\_RECONSTRUCTION\_PROCESSOR\_TYPE\_CPU = 1,

NUI\_FUSION\_RECONSTRUCTION\_PROCESSOR\_TYPE\_AMP = 2

} NUI\_FUSION\_RECONSTRUCTION\_PROCESSOR\_TYPE;

NUI\_FUSION\_RECONSTRUCTION\_PROCESSOR\_TYPE\_CPU 使用CPU重建

NUI\_FUSION\_RECONSTRUCTION\_PROCESSOR\_TYPE\_AMP 使用GPU重建